HUMAN ACTION RECOGNITION BY EMBEDDING SILHOUETTES AND VISUAL WORDS

BEHROUZ SAGHAFI KHADEM

School of Computer Engineering

A thesis submitted to the Nanyang Technological University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

2013
Acknowledgments

I would like to express my sincere thanks to my supervisor, Dr. Deepu Rajan for his invaluable guidance throughout my PhD program. I regard myself fortunate to have him as my advisor in my research study. His insight has helped me to choose the topic of my research. His involvement and constant encouragement motivates me to pursue my research passionately. I would also like to appreciate the Center for Multimedia and Network Technology (CeMNet) for providing me with resources and also School of Computer Engineering (SCE) and Nanyang Technological University (NTU) for offering me the scholarship to pursue my Ph.D. candidate study. Last but not least, I would like to thank my family for their continuous support and encouragement and also my wife, Elahe who has always stood by me throughout my PhD program.
## Contents

Acknowledgments .......................................................... i  
List of Figures ............................................................ vi  
List of Tables ............................................................... x  
Acronyms ................................................................. xi  
Abstract ................................................................. xii

1 Introduction .............................................................. 1  
1.1 Motivations ......................................................... 2  
1.2 Challenges .......................................................... 4  
1.3 Objectives ............................................................ 5  
1.4 Contributions ....................................................... 6  
1.5 Outline ............................................................... 7

2 Literature Review ........................................................ 9  
2.1 Features used in Action Recognition ............................... 9  
  2.1.1 Optical Flow ...................................................... 9  
  2.1.2 Point Trajectories .............................................. 10  
  2.1.3 Silhouettes ...................................................... 10  
  2.1.4 Spatio-temporal Interest Points ............................... 11  
2.2 Action Recognition Approaches .................................... 11  
  2.2.1 Sequential Approaches ........................................ 12  
  2.2.2 Volumetric Approaches ....................................... 17  
2.3 Conclusions ........................................................ 21
3 Human Action Recognition using Pose-based Discriminant Embedding 23

3.1 Introduction .................................................. 23
   3.1.1 Motivation and Overview of Approach ..................... 25
   3.1.2 Organization .............................................. 27
3.2 Related Work .................................................. 27
3.3 Pose-based Discriminant Embedding .......................... 29
   3.3.1 Distances between sets of points ......................... 30
   3.3.2 Optimal Embedding Computation .......................... 31
3.4 Preprocessing ................................................ 35
   3.4.1 Period Estimation ........................................ 35
   3.4.2 Warping .................................................. 37
   3.4.3 Aligning ................................................. 38
3.5 Experimental Results ........................................ 39
   3.5.1 Experiments on Weizmann Database ........................ 39
   3.5.2 Experiments on Maryland Database ........................ 51
   3.5.3 Experiments on KTH Database ............................ 53
3.6 Conclusion .................................................... 54

4 Learning Semantic Visual Codebook for Action Recognition by Embedding into Concept Space 56

4.1 Introduction .................................................. 56
   4.1.1 Overview of the proposed framework ..................... 57
4.2 Related Work ................................................ 58
4.3 Concept Space ............................................... 59
   4.3.1 Embedding into concept space using Latent Semantic Analysis 60
   4.3.2 Embedding into concept space using Probabilistic Latent Semantic Analysis 61
   4.3.3 Embedding into concept space using Canonical Correlation Analysis 63
4.4 Feature Extraction ......................................... 66
4.5 Experiments ................................................. 67
   4.5.1 Experiments using Latent Semantic Space ................ 67
4.5.2 Experiments using Canonical Correlation Space .......................... 70
4.5.3 Comparison with Other Methods ........................................... 72
4.6 Conclusion ............................................................................. 73

5 Efficient Combination of View-dependent Histograms for Multi-view
Human Action Recognition .............................................................. 74
5.1 Introduction ............................................................................ 75
5.1.1 Overview of Approach ........................................................ 78
5.1.2 Organization .................................................................... 78
5.2 Related works ...................................................................... 80
5.3 Histogram Representation of Video .......................................... 81
5.3.1 Separable Linear Filters ....................................................... 81
5.3.2 Space-Time Corner Detector ............................................... 82
5.3.3 Codebook Size .................................................................. 83
5.4 Efficient Combination of View-dependent Histograms ............... 83
5.4.1 Computing Histogram Intersection Kernel ......................... 84
5.4.2 Computing Radial Basis Function Kernel with $\chi^2$ Distance ... 84
5.4.3 Learning an Efficient Combination of Kernels ..................... 84
5.5 Experimental Results .............................................................. 86
5.5.1 IXMAS Multi-view Action Dataset ....................................... 86
5.5.2 Results and Analysis .......................................................... 88
5.5.3 Viewpoint Analysis ............................................................. 88
5.5.4 Effect of Feature Type ......................................................... 90
5.5.5 Effect of using more Codebooks ......................................... 91
5.5.6 Effect of Kernel Type .......................................................... 92
5.5.7 Comparison of Different Fusion Methods ............................. 93
5.5.8 Comparison with Other Methods on IXMAS Dataset ............ 94
5.6 Conclusion ............................................................................. 94

6 Conclusions ............................................................................... 96
6.1 Summary of Contributions ...................................................... 96
6.2 Future Research .................................................................... 97
List of Figures

2.1 Taxonomy of human action recognition techniques. This dissertation discusses manifold learning and histogram based approaches. 12

2.2 Using Fourier transform to represent sequence of silhouettes. Figure taken from [1]. 15

2.3 Temporal templates; (a) key frame (b) motion energy image (MEI) (c) motion history image (MHI). Figure taken from [2]. 17

2.4 The 3D space-time objects pertaining to jumping jack, walking and running. Figure taken from [3]. 18

2.5 Spatio-temporal features detected for walking: (a) 3D plot of spatio-temporal motion pertaining to leg and associated features; (b) Features overlaid on frames. Figure taken from [4]. 20

3.1 Examples of postures for run and its trajectory in a possible action space. 25

3.2 Training phase of the proposed method. 26

3.3 Testing phase of the proposed method. 26

3.4 Trajectories of one period of action run with the same duration for 2 different persons (red and black). Arrows show the deviations due to appearance change. 34

3.5 Period estimation for walk sequence pertaining to person 1 (Daria) of Weizmann database [3] with a length of 84, (a) similarity $S$, (b) $\hat{z}$, the middle column vector $z$ of $S$ which is linearly detrended, (c) its autocorrelation. 36

3.6 False peak detections by zero-derivative method indicated by red vertical lines. 36
3.7 Finding $T$ interpolation time instances for a $P$ length sequence. On the time axis, the green circle markers show the time instances for existing frames $(1, 2, \ldots, P)$ and the blue triangle markers show the interpolation time samples. Real poses from one period of the action run is shown above the time axis and the interpolated poses are represented below the axis.  

3.8 Aligning two warped sequences of action run: The first pose of sequence 1, is similar to the $5^{th}$ pose of sequence 2. The other poses are aligned in a cyclic manner. For this example: $b = 4$.  

3.9 Recognition accuracy versus dimension($l$), for different values of $T$ for SCD as the distance metric.  

3.10 Studying the effect of $T$: comparing the maximum and mean of recognition rate in the stable interval from the plots in figure 3.9 for different values of $T$.  

3.11 Recognition accuracy versus dimension($l$), for different values of $T$ for MHD as the distance metric.  

3.12 Studying the effect of $T$: comparing the maximum and mean of recognition rate in the stable interval from the plots in figure 3.11 for different values of $T$.  

3.13 Recognition accuracy versus dimension, using the test sequences without warping for $T = 6$ and MHD as distance.  

3.14 3D visualization of action points: (a) LDA, (b) PCA, (c) SLPP, (d) PDE.  

3.15 Results of Robustness to noise.  

3.16 Sample images of Weizmann’s robustness database for deformations [3]. From left to right and from top to bottom: swinging bag, carrying briefcase, knees up, limping man, sleepwalking, occluded legs, walking in a skirt, walking with a dog, presence of a pole, normal walk, respectively.  

3.17 Sample images of Weizmann’s robustness database for viewpoint [3]. From left to right and from top to bottom: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, respectively.  

3.18 Examples of silhouettes of Maryland database [5]. From left to right: pick up object, jog in place, push, squash, wave, kick, bend to the side, throw, turn around, and talk on cell phone, respectively.
3.19 Examples of computed edge maps for in-place actions of KTH dataset. 54
4.1 Constructing the semantic visual vocabulary. 58
4.2 Training steps of the proposed method. 58
4.3 Constructing the semantic visual vocabulary. 66
4.4 Some examples of detected interest points for six actions of KTH dataset overlaid on sample images. The patches indicate extracted cuboids in sequences. As seen interest points are mainly extracted from parts which are involved in the main motion. 68
4.5 Performance of proposed method on KTH action dataset with different number of topics using (a) LSA (b) pLSA for embedding. 69
4.6 KTH action dataset. (a)Comparison of results with the classic framework for different sizes of vocabulary. (b)Confusion matrix for the best result achieved. 69
4.7 Effect of changing the dimension of CCA space. 71
4.8 Effect of changing the vocabulary size. 71
4.9 Confusion matrix for the best result on KTH dataset. 72
5.1 Example images from actions cross arms and check watch from side view as well as top view. 77
5.2 The method for generating basic kernels $K_{c,f,v,k}$ from all the possible combinations of constitutive factors, namely, camera viewpoint ($c \in \{\text{View 1, ..., View C}\}$), feature type ($f \in \{\text{Separable Linear Filters, Space-time Corner Detector}\}$), codebook size ($v \in \{V, 2V\}$) and kernel type ($k \in \{\text{HIK distance in a linear function, Chi-Square distance in an exponential function}\}$). The basic kernel $K_{c,f,v,k}$ is weighted with $w_{c,f,v,k}$. $K$, the final kernel used, is the linear combination of basic kernels. 79
5.3 Example images of the actions in IXMAS dataset. 87
5.4 Example views from five cameras in IXMAS dataset for kick action. 88
5.5 Confusion matrix for the best result achieved on IXMAS. 89
5.6 Accuracy for each view (camera) of IXMAS. 89
5.7 Best accuracy for combination of views in IXMAS. 90
5.8 Performance of each feature type. ........................................ 91
5.9 Performance of using more codebooks. .............................. 91
5.10 Performance of each kernel type and combination of them. .... 92
5.11 Comparison of different fusion methods. ............................ 93
## List of Tables

3.1 Comparison of different dimension reduction methods. The number in parenthesis shows the optimum dimension which is found empirically. $T = 6$ is used for PDE. The experiment is done using a single action cycle. . . 44

3.2 Comparison with recent results on Weizmann dataset. . . . . . . . . . . . 46

3.3 Results of Deformation robustness test. . . . . . . . . . . . . . . . . . . . 50

3.4 Results of viewpoint robustness test. . . . . . . . . . . . . . . . . . . . . 51

3.5 Comparison of different dimension reduction methods. The number in parenthesis shows the optimum dimension. $T = 6$ is used for PDE. The experiment is done using a single action cycle. . . . . . . . . . . . . . . . 52

3.6 Comparison with other methods on KTH dataset for the in-place actions. 54

4.1 Comparison with recently reported results for KTH dataset. . . . . . . . 73

5.1 Comparison of Recognition Accuracy on IXMAS dataset. . . . . . . . 94
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOW</td>
<td>Bag Of Words</td>
</tr>
<tr>
<td>CCA</td>
<td>Canonical Correlation Analysis</td>
</tr>
<tr>
<td>DTW</td>
<td>Dynamic Time Warping</td>
</tr>
<tr>
<td>DM</td>
<td>Diffusion Map</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation Maximization</td>
</tr>
<tr>
<td>HCI</td>
<td>Human-Computer Interaction</td>
</tr>
<tr>
<td>HIK</td>
<td>Histogram Intersection Kernel</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>HOF</td>
<td>Histogram of Optical Flow</td>
</tr>
<tr>
<td>HOG</td>
<td>Histograms of Oriented Gradients</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear Discriminant Analysis</td>
</tr>
<tr>
<td>LDS</td>
<td>Linear Dynamical Systems</td>
</tr>
<tr>
<td>LE</td>
<td>Laplacian Eigenmaps</td>
</tr>
<tr>
<td>LLE</td>
<td>Locally Linear Embedding</td>
</tr>
<tr>
<td>LPP</td>
<td>Locality Preserving Projections</td>
</tr>
<tr>
<td>LSA</td>
<td>Latent Semantic Analysis</td>
</tr>
<tr>
<td>LSTDE</td>
<td>Local Spatio-Temporal Discriminant Embedding</td>
</tr>
<tr>
<td>MACH</td>
<td>Maximum Average Correlation Height</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A Posteriori Probability</td>
</tr>
<tr>
<td>MEI</td>
<td>Motion Energy Image</td>
</tr>
<tr>
<td>MHD</td>
<td>Median Hausdorff Distance</td>
</tr>
<tr>
<td>MHI</td>
<td>Motion History Image</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum Likelihood Estimation</td>
</tr>
<tr>
<td>MMI</td>
<td>Maximization of Mutual Information</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>PDE</td>
<td>Pose-based Discriminant Embedding</td>
</tr>
<tr>
<td>PLSA</td>
<td>Probabilistic Latent Semantic Analysis</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>SCD</td>
<td>Spatio-temporal Correlation Distance</td>
</tr>
<tr>
<td>SLPP</td>
<td>Supervised Locality Preserving Projections</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
</tbody>
</table>
Abstract

With the availability of cheap video recording devices, fast internet access and huge storage spaces, the corpus of video that is accessible has grown tremendously over the last few years. Processing of these videos to achieve end-user tasks such as video retrieval, human-computer interaction (HCI), biometrics etc. require automatic understanding of content in the video. Human action recognition is one aspect of video understanding that is useful in surveillance, behavioral analysis and HCI. Although this problem has been studied for quite some years now, challenges still exist in terms of cluttered background, intra-class variance and inter-class similarity, occlusion etc. In this thesis, we propose three methods for action recognition.

First, we propose a novel embedding for learning the manifold of human actions which is optimum based on spatio-temporal correlation distance (SCD) between sequences. Sequences of actions can be compared based on distances between frames. However comparison based on between-sequence distance is more efficient and effective. In particular, our proposed embedding minimizes sum of distances between intra-class sequences while maximizing sum of distances between inter-class points. Action sequences are represented by key postures chosen equidistantly from a semantic period of action. The projected sequences are compared based on SCD or Hausdorff distance in a nearest neighbor framework. The method not only outperforms other dimension reduction methods but is comparable to the state of the art on three public datasets. Moreover it is robust to additive noise, occlusion, shape deformation and change in view point up to a large extent.

Second, we propose an approach for introducing semantic relations into the bag-of-words framework for recognizing human actions. In the standard bag-of-words framework, the features are clustered based on their appearances and not their semantic relations. We exploit Latent Semantic Models such as LSA and pLSA as well as Canonical
Correlation Analysis to find a subspace in which visual words are more semantically distributed. We project the visual words into the computed space and apply k-means to obtain semantically meaningful clusters and use them as the semantic visual vocabulary which leads to more discriminative histograms for recognizing actions. Our proposed method gives promising results on the challenging KTH action dataset.

Finally, we introduce a novel method for combining information from multiple viewpoints. Spatio-temporal features are extracted from each viewpoint and used in a bag-of-words framework. Two codebooks with different sizes are used to form the histograms. The similarity between computed histograms are captured by HIK kernel as well as RBF kernel with $\chi^2$ distance. Obtained kernels are linearly combined using proper weights which are learned through an optimization process. For more efficiency, a separate set of optimum weights are calculated for each binary SVM classifier. Our proposed method not only enables us to combine multiple views efficiently but also models the action in multiple spaces using the same features, thereby increasing performance. Several experiments are performed to show the efficiency of the framework as well as the constitutive parts. We have obtained the state of the art accuracy of 95.8% on the challenging IXMAS multi-view dataset.
Chapter 1

Introduction

The human visual system has an incredible capability in fast and accurate analysis and interpretation of visual data. We are able not only to detect and recognize static objects, but also to distinguish patterns of motion and temporal events in complex environments. Even a child can simply specify whether a person is walking, running or fighting. Over the last few decades, the empowerment of machines with the ability to interpret visual data has seen tremendous progress, yet the state-of-the-art is nowhere near what the human visual system can achieve.

The corpus of video data is growing rapidly. Almost all of the commonly used consumer hardware such as laptops, cellphones or digital cameras are able to record videos. Thanks to video sharing and social networking websites a huge amount of video data is being uploaded and shared everyday. In fact, every minute, 72 hours of video are being uploaded to YouTube [6]. This enormous amount of data requires smart tools for efficient searching and navigation. Currently the searching and retrieval is based on textual tags. However issues like improper tagging or different languages usually lead to retrieval of unrelated material. Moreover some data are inexpressible by words. All these suggest that we use the more natural search using content which requires tools for understanding video data and interpreting it. With the increase in surveillance systems the need for automatic monitoring will grow. For instance a march 2011 statistics reveals that there are a total of 1.85 million CCTV cameras in UK. This means that for every 32 people in UK, there is one CCTV camera operating [7]. The online processing of this huge amount of data stream is impossible for human labor. Thus automatic interpretation of video is an alternative.
The primary focus in this thesis is the automatic recognition of human action categories in sequences of video. In particular, given videos containing a human performing an action, we are interested in classifying the action into one of the known categories or classes (e.g., running, clapping, or boxing) which have been learned. Recognizing the behavior of humans in sequences of video can aid in a wide variety of applications including content-base video analysis, surveillance, human-computer interaction, gait recognition and virtual action synthesis. It is challenging due to the large variances that exist within one action category. For instance, the actions can be done by different persons with diverse appearances performing the action with various styles. Also, actions can happen under various conditions such as different illuminations, backgrounds, viewpoints, or rates.

The terms action and activity have been used interchangeably in the literature. To clarify their meanings, we refer to [8] that defines action as simple patterns of motion performed by a single person which typically lasts for a short duration. Examples of actions are bending, running, waving, and throwing. Activity on the other hand, is referred to a sequence of actions performed by several humans in which the subjects may interact with each other. Activities are typically done in much longer durations compared to actions. Examples are two persons dancing tango, a basketball team scoring a goal, or a bank attack. In our work, we concentrate on actions rather than activities. Nevertheless, actions can be considered as the building blocks of activities.

The research we pursue in this thesis is introduced in this chapter. We begin by discussing the motivations. Then we carry on by describing the challenges we face. Later on, the objective is being summarized. Subsequently the main contributions are highlighted and finally the outline of the thesis is presented.

1.1 Motivations

Apart from scientific curiosity, our motivation comes from the wide variety of applications which could benefit from this computer vision system. Some of the more important ones are [8, 9]:

**Content-based Video Retrieval and Summarization.** Video sharing websites like YouTube, Google Video, Vimeo and Dailymotion require more efficient algorithms for
searching and retrieval of the video content. Although searching based on text may be faster in a collection of millions of videos, they may not always retrieve videos that are related to the queried video. A natural alternative is to analyze the multimedia content of video and to retrieve based on the underlying semantics. Using techniques for human action recognition can be beneficial towards decoding the contents and annotation of videos. Similarly summarization is another important application especially for contents like sport events or news.

** Behavioral Intelligence.** The majority part of the security and surveillance systems operating in airports, banks, shopping malls or hospitals rely on video cameras, which traditionally needs human operators to detect any abnormal activities. By increasing use of these systems, computer-based solutions for monitoring to replace or help human operators are required.

** Human-computer Interaction.** Most machines receive inputs via interfaces such as keyboard, mouse and touch screen. Although these devices can interact with high precision and lower cost, they are less natural and expressive compared to using gestures. Using a magnet [10] or infrared LED [11] may not be always convenient and pleasant and may constrain movement. Thus the more natural and appealing way is to rely on absolute visual information and try to recognize hand and body gestures captured by the camera. So human action recognition can play a significant role towards more natural human-machine interactions.

** Gait Recognition.** Biometrics deals with the algorithms for recognition of human subjects from their physical or behavioral cues. The former includes finger prints, face, iris etc. Recognizing subjects based on physical biometrics need cooperation of subjects and cannot be done in some cases. So behavioral biometrics have gained much attention recently which studies the recognition of humans from their behavior or style of doing their activities. The advantage of these methods is that subject’s cooperation is not necessary and it can proceed without interruption or interfere with the subject’s activity. Gait recognition is one of the most important applications of behavioral biometrics which uses the same approaches as recognition of actions.

** Virtual Action Synthesis.** Synthesizing realistic humans or human motions in a virtual environment has wide applications in game industry as well as the new animation
movies. So any algorithm in modeling the human motion can help to advance the industry in these fields.

1.2 Challenges

Despite great attempts in the computer vision research, unconstrained recognition of human actions in real-world conditions remains unsolved to a great extent. This problem is challenging since a computer-based recognition system, learned on a limited training set, has to recognize an action performed by an unknown person in a cluttered background with unfavorable recording setting. Here we discuss the main challenges we face in recognition of human actions:

**Intra-class Variance and Inter-class Similarity.** Development of an automatic action recognition system is a formidable challenge, partly due to the possible variations of an action belonging to the same category. Different subjects can perform the same action. Their different appearance or clothing change the features extracted from the actions, leading to errors. Moreover the style with which a person executes an action may vary, for instance there are different patterns and stride lengths for walking. An ideal action recognition system should generalize over variances of a class in order to recognize a previously unseen instance.

Moreover there may be significant similarity between patterns of two different classes of actions. For example walking, jogging and running all have almost the same patterns. Also check watch and cross arms share the same movements partly. A good action recognition system should be sufficiently discriminant across different categories.

**Cluttered or Complex Background.** Cluttered background may add up noise and unrelated details to the system which can lead to errors in the recognition. Point trajectories are usually difficult to track in cluttered backgrounds. Also in the cases where silhouette features are required, complex backgrounds makes it hard to extract reliable foregrounds.

**Illumination Change.** The illumination conditions in the video can change due to various factors including different angles of emitted light, multiple sources of lighting, reflection from objects and camera saturation. The change in the lighting conditions can change the extracted features dramatically, thus makes it hard to recognize accurately.
**Viewpoint Change.** The change in the camera viewpoint will change the image observation due to camera perspective effects. This will result in substantial changes in motion and structure features. A good action recognition system should be robust to viewpoint change to some extent.

**Occlusion.** Occlusions can be caused by the subject itself (self-occlusions), by other objects or due to being outside the field of view. They lead to degradation of visual appearance of parts of the subject. These parts may be essential or discriminative in recognizing an action.

**Execution Rate Variance.** There can be significant variation in the rate in which the action is being executed. This can affect the action recognition especially when the used features are based on motion or temporal windows. A robust action recognition system should be invariant to rates of performance of an action.

The availability of standard databases for action recognition has enabled an objective evaluation of the various algorithms that have been proposed. Some of these databases include Weizmann dataset [3], Maryland dataset [5], KTH dataset [4], INRIA IXMAS dataset [12], UCF dataset [13] and Hollywood dataset [14]. All these datasets include intra-class variance and inter-class similarity among actions. Moreover, Weizmann dataset targets viewpoint change and occlusion. Maryland dataset evaluates the execution rate variance. There is considerable change in the illumination as well as camera zooming in KTH dataset. INRIA IXMAS dataset is mostly designed to test the viewpoint change. UCF contains substantial variation in illumination as well as viewpoint. Finally Hollywood dataset includes a mixture of all the variation discussed so far. Given the challenging nature of some of these datasets, there is evidently a long way to go before very high recognition rates can be achieved on these datasets.

### 1.3 Objectives

Our main objectives in this dissertation are three fold:

- To develop an algorithm to handle intra-class variance and inter-class similarity in human actions.
Chapter 1. Introduction

- To develop an approach for considering semantic relations in bag-of-words framework by computing a concept space in which visual words are more semantically distributed.

- To develop an algorithm for multi-view action recognition which helps in overcoming occlusion.

1.4 Contributions

The main contributions of this dissertation are as follows:

- We propose a novel embedding to address intra-class variance and inter-class similarity in human actions which is optimum in the sequence recognition framework based on Spatio-temporal Correlation Distance as the measure of distance. Specifically, the proposed embedding minimizes the sum of the distances between intra-class sequences while seeking to maximize the sum of distances between inter-class points. Action sequences are represented by key poses chosen equidistantly from one action period. The action period is computed by a modified correlation-based method. Action recognition is achieved by comparing the projected sequences in the low-dimensional subspace using SCD or Hausdorff Distance in a nearest neighbor framework. Several experiments are carried out on three popular datasets. The method is shown not only to classify the actions efficiently obtaining results comparable to the state of the art on all datasets, but also to be robust to additive noise and tolerant to occlusion, deformation and change in viewpoint. Moreover, the method outperforms other classical dimension reduction techniques and performs faster by choosing less number of postures.

- A novel approach is proposed for introducing semantic relations into the bag-of-words framework for recognizing human actions. Latent semantic models such as LSA and pLSA as well as Canonical Correlation Analysis has been used to find a subspace in which the words are more semantically distributed. The semantic features using LSA is derived from factorization of word-video occurrence matrix by SVD. Similarly using pLSA topic-specific distributions of words infer the
dimensions of concept space. Moreover applying Canonical Correlation Analysis between original visual features and their video co-occurrence representation will give semantic features. We apply k-means clustering in the computed space to find semantically meaningful clusters and use them as the semantic visual vocabulary. Incorporating the semantic visual vocabulary the features are quantized to form more discriminative histograms. Eventually the histograms are classified using an SVM classifier. We have tested our approach on KTH action dataset and achieved promising results.

- Occlusion can be handled by using multiple cameras from different viewpoints. We introduce a new method to efficiently combine the data from multiple views. Spatio-temporal features are extracted from each viewpoint and used in a bag-of-words framework to form histograms. Two different sizes of codebook is exploited. The similarity between obtained histograms are represented via Histogram Intersection kernel as well as RBF kernel with $\chi^2$ distance. Finally we combine all the basic kernels generated by different viewpoints as well as different feature types, codebook sizes and kernel types. The final kernel is a linear combination of basic kernels which have been properly weighted based on an optimization process. For higher accuracy the set of kernel weights are being computed separately for each binary SVM classifier. Our method not only enable us to efficiently combine the information from multiple viewpoints but by modeling the same features in different spaces improves the performance. Several experiments have been carried out to verify the efficiency of the proposed method as well as the constitutive parts. We have achieved the state of the art accuracy of 95.8% on the challenging multi-view IXMAS dataset.

1.5 Outline

This thesis consists of six chapters. Chapter 2 discusses the related work in human action recognition considering different features and representation methods. Chapter 3 details our novel embedding method using silhouettes with several experiments and discussions.
Chapter 1. Introduction

Chapter 4 presents our novel approaches for introducing semantic relations into bag-of-words framework using spatio-temporal features. Subsequently, chapter 5 introduces our proposed method for combination of multiple views. Finally we conclude the work in chapter 6 and give directions for future research.
Chapter 2

Literature Review

There is a rich literature on action recognition thanks to its long history in computer vision research. Consequently, there have been several survey papers on the topic including Cedras and Shah [15], Gavrila [16], Aggarwal and Cai [17], Moeslund and Granum [18], Wang et al. [19], Poppe [20], Kruger et al. [21], Turaga et al. [8], Poppe [9] and Aggarwal and Ryoo [22]. In this chapter we review the more important works that are related to this dissertation. Approaches to human action recognition extract informative features from the input video. These features are used to model the actions with some sort of representations and finally recognize the action categories. Correspondingly we start with reviewing the common features which are used for action recognition. Subsequently we explore the approaches to modeling and recognizing actions. The taxonomy is broadly based on [8] and [22].

2.1 Features used in Action Recognition

Videos consist of massive amount of information in the form of spatio-temporal pixel intensities. Not all of the information is useful in the task of recognizing actions. Hence, selecting useful information in the form of essential features is critical. Here we briefly describe some important and common features used in the task of action recognition.

2.1.1 Optical Flow

Optical flow is defined as the motion vector of each individual pixel in the image. It is an appropriate tool for describing the moving regions [8]. Beauchemin and Barron
Chapter 2. Literature Review

[23] provide a comprehensive survey on computation techniques of optical flow. These techniques assume that moving areas have the same color or intensity. Optical flow describes both the regions undergoing motion as well as the velocity of motion. Computation of optical flow is not robust to noise and illumination changes [8]. Also when we are dealing with videos with low quality or non-smooth motion, the computed vectors for optical flow are not accurate. Efros et al. [24] have introduced a new feature based on optical flow which is used for low resolution videos. In their novel feature they smooth and aggregate the noisy patterns of optical flow to form an informative descriptor.

2.1.2 Point Trajectories

Trajectories of important points on the subject have been used as a representation for actions. The work of Ali et al. [25] is an example of this kind of representation which uses chaos theory in classifying time series resulting from point trajectories. Finding these trajectories need accurate tracking of important joints. Hence, fast moving subjects, occlusions and cluttered or noisy backgrounds may cause difficulties [8].

2.1.3 Silhouettes

Silhouettes are important and informative features in recognizing actions and have been used widely [3, 26–30]. One of the common methods for extracting silhouettes is background subtraction [31]. It is a popular method for segmenting the foreground moving regions from the background by constructing a model for scene background and comparing each frame to it. In simple and homogeneous backgrounds, silhouettes are extracted by background subtraction. But for more complicated backgrounds or for moving camera, extracting correct foreground is not easy.

In order to obtain shape information from silhouettes, they should be described using appropriate descriptors. Several methods based on regions, boundary, skeleton and bounding box have been used to encode the shape from extracted blobs. Moments [32] have been proposed to describe the region. Moreover the contour of the blob has been described by chain codes [33] or landmark-based descriptors [34]. Medial axis transform skeleton is another method for representing the shape information using skeleton curves. Furthermore the pixels within the bounding box has been used to model the shape.
These pixels are vectorized and embedded into proper subspaces using dimension reduction techniques [28–30]. This method of description have the benefit of robustness with respect to boundary noise. All these methods describe individual silhouettes from each frame. There are also methods which use all the silhouettes of a sequence to construct a volume and extract features from the entire space-time shape [3].

### 2.1.4 Spatio-temporal Interest Points

There are other features that can be categorized generally as spatio-temporal interest points. These features come from gradients in the domain of space or time, where local extrema become the interest points. The high responses of gradient function in the space domain indicate corners and in the time domain they infer motion. Schuldt et al. [4] generalized the Harris corner detector to the spatio-temporal case. Dollar et al. [35] stated that direct 3D counterparts to commonly used 2D interest points detectors are inadequate, because the time domain is intrinsically different from spatial domain. Instead, their proposed interest points are detected based on a response function using Gaussian kernel in space and Gabor function in time. These methods are fast and easy to apply because they are based on simple convolution operations. They are usually helpful in scenarios where we have complicated backgrounds or low resolution quality which the previous methods in feature detection face difficulties [8]. Spatio-temporal interest points describe local regions, but ignore the global structural information which might be useful for recognition.

### 2.2 Action Recognition Approaches

Figure 2.1 shows the tree-based taxonomy of approaches we use throughout this chapter. All methodologies for action recognition can be broadly classified into two categories: sequential approaches and volumetric approaches. Sequential approaches model human action as a sequence of observations which typically refer to individual frames. They represent each observation by a separate feature vector. On the other hand volumetric approaches treat action as a 3D volume across space-time dimensions. This volume is formed by concatenation of frames along time axis. A set of features are extracted from
Figure 2.1: Taxonomy of human action recognition techniques. This dissertation discusses manifold learning and histogram based approaches.

2.2.1 Sequential Approaches

Sequential approaches convert a sequence of frames into a sequence of features. The likelihood between the sequence and the learned action categories is measured. The query sequence is recognized as the category of the sequence with the highest likelihood. We further classify the sequential approaches into two categories: exemplar-based approaches and state-space approaches. Exemplar-based approaches describe the action categories by the training samples or their representations. However state-based approaches construct a model for each category and compare the models against each other.

2.2.1.1 Exemplar-based Approaches

Exemplar-based approaches represent human actions by templates of sequences which is obtained from the training data. The similarity of the new sequence to the template sequences are measured and used to determine the category of query. If the sequence of features is used as it is without embedding the features onto another subspace we
are dealing with non-embedding methods. However if an embedding is used in order to
construct the templates, the method is categorized as a manifold learning approach.

**Non-Embedding Methods.** Non-embedding methods do not apply any mapping
on the obtained sequence of features. Instead they rely on the measure of distance to
account for intra-class variance or other flexibilities. Actions can be performed with
different rates or with variant styles. The distance measure between sequences should
consider these variations. One of the popular distances which is robust to these kinds of
variances is Dynamic Time Warping (DTW) which is originally used for speech recogni-
tion [36]. An optimum nonlinear match is obtained by the DTW algorithm. DTW is
robust to speed variations. Darrell and Pentland [36] proposed a DTW-based method for
gesture recognition. Their system stores several models of an object in different status.
They refer to these models as views. The correlation between frames and each view-model
is computed as a function of time for each query video. The gesture templates used are
the means and variances of this function computed for the training videos. DTW is used
to match the query video to the gesture templates.

Efros et. al [24] proposed a method for recognizing actions at a distance like ballet
movements or soccer plays. The humans present in these videos typically have a low
resolution. Temporal difference is used to track humans and subsequently optical flow is
computed at each frame. They convert the optical flow to a motion descriptor denoted
as blurry motion channel. A frame to frame similarity based on nearest neighbor is cal-
culated for two sequences to form a similarity matrix. Diagonal patterns in the similarity
matrix is used for recognition.

Veeraraghavan et al. [5] have developed an extension of DTW algorithm. Knowing
that actions can be performed with partly different execution rates, they use two functions
for modeling the action execution: a function space of all possible time warping and a
function of changes in the features over time. These nonlinear models are obtained for
each action category and used when matching two sequences.

Deformable part model has been used by Xie et al. [37] for both human detection and
pose estimation which are finally integrated into action recognition task. Histograms of
oriented gradients (HOG) feature pyramid has been adopted to represent features. One
of the advantages of their method is that they do not require satisfactory foreground
extraction.
Hoai et al. [38] use multi-class SVM to train a model capable of doing segmentation and action recognition simultaneously using dynamic programming.

Maji et al. [39] propose a framework based on poselet which is a supervised body part detector. They present a distributed representation of pose and appearance of human denoted as poselet activation vector. This framework is not only capable of pose estimation but combined with other sources of information like interactions with other objects/people can be used for action recognition. The recognition is performed on people’s bounding boxes.

**Manifold Learning Methods.** Although non-embedding methods can handle speed variations by using distances like DTW, they are not able to consider other complex intra-class variations. Thus it may be necessary to embed features into spaces in which the intra-class variation is minimum while different categories are easily distinguishable. These methods which apply some sort of mapping are denoted as manifold learning methods. They typically use silhouettes as their main features. Silhouettes are considered as points in the high dimensional image space. It is assumed that these points lie on a lower-dimensional manifold which represent the action. The action manifold can be learned via the classical dimension reduction methods like Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) or any other embedding.

Wang and Suter [28] have proposed a manifold learning framework based on Locality Preserving Projections (LPP). After embedding into lower dimensional space, the action trajectories are compared by using two distance metrics denoted as Spatio-temporal Correlation Distance (SCD) and Median Hausdorff Distance (MHD). Finally the actions are classified using a nearest neighbor classifier. In another attempt Jia and Yeung [30] proposed an embedding which is discriminative in both spatial and temporal aspects. In their embedding the spatial criteria is the same as Fisher’s LDA. However, for the temporal fold, the principal angles between those temporal subspaces associated with inter-class points are being maximized. Masoud and Papanikolopoulos [40] have used a framework similar to [28] but using PCA on IIR filtered images. In a different work, Wang et al. [1] use Discrete Fourier Transform as well as Discrete Wavelet Transform to compute shape features (Fig. 2.2) and apply kernel LPP to learn the underlying action manifold. Wu et al. [41] in an online recognition scheme, use incremental discriminant
analysis of canonical correlations to gradually learn the change in appearance instead of learning the whole training set in advance.

### 2.2.1.2 State-space Approaches

State-space approaches obtain a statistical model for each human action composed of a set of states. For each model the probability of generating the observed sequence is computed. This probability is the likelihood between the query image sequence and the model. The likelihood is measured by either maximum likelihood estimation (MLE) or maximum a posteriori probability (MAP). The exemplar-based approaches are more appropriate for simple actions. However state-space approaches are used for more complicated actions such as steps in a ballet, juggling a ball or conducting an orchestra using complex hand gestures [8]. One of the drawbacks of state-space approaches compared to exemplar-based ones is that they require a large set of training data as the actions become more complex [22]. State-space approaches are further classified based on the space of states. Hidden Markov models (HMM)s are used when the state space is constrained to be a finite set of symbols. Otherwise dynamical systems are being applied.

**Hidden Markov Models.** HMMs are one of the most popular state-space approaches. HMM in the discrete format consists of a finite set of discrete states, in which the system transits with a known probability called the transition probability. HMMs first gained popularity in speech recognition. One of the most complete reviews on HMM is the tutorial from Rabiner [42]. Following the wide applicability in speech recognition, researchers in computer vision also started using them. One of the first works on action recognition using HMM was the work of Yamato et al. [43]. In their work they trained an HMM based on a sequence of background subtracted images and apply it to classify a database of tennis shots. HMM have also been used in gesture recognition applications.
Chapter 2. Literature Review

[44]. Tung and Matsuyama [45] have proposed a topology-based dictionary based on Reeb graphs to describe 3D video sequences in a Markovian framework. They use this encoding for both video summarization and 3D action recognition.

Many action recognition tasks can be broken down into multiple views such as body postures and hand shapes which interact with each other. In order to efficiently capture these interactions, Song et al. [46] proposed multi-view latent variable discriminative models capable of learning both the shared and specific substructures. They suggest three different topologies- linked, coupled, and link-coupled- which differ on the way the substructures are modeled.

Although HMM is an appropriate tool for recognizing actions with unknown start and end times, their applicability is restricted to stationary patterns due to assumptions in the Markovian model [8].

**Dynamical Systems.** Linear dynamical systems (LDS) are generalization of HMM when we have continuous values instead of finite discrete states. The simplest form of LDS is the Gauss-Markov process. This is actually a continuous state-space generalization of HMM with a Gaussian model for observations. [47, 48] are two applications of LDS in action recognition. Like HMMs, LDS is also only applicable to stationary actions, so some researchers have also used nonlinear dynamical systems [49]. Although they can handle non-stationary actions, learning in these systems is complicated and usually needs large amount of training data [8]. Most of the methods based on dynamical systems are limited to applying features defined in Euclidean spaces like joint trajectories. Chaudhry et al. [50] have proposed a generalization of Binet-Cauchy kernels to nonlinear systems in which their inputs can be in non-Euclidean spaces e.g. histograms. This is an example of using spatio-temporal features in a dynamical system framework.

In order to handle large spatio-temporal variations within action classes, Wang et al. [51] have introduced a novel component denoted as substructure transition model in which the sparse and global temporal transitions are encoded. The other novel component proposed is the discriminative boundary model in order to more accurately locate the transition boundaries between actions.
Chapter 2. Literature Review

2.2.2 Volumetric Approaches

As opposed to sequential approaches which treat action as a sequence of observations, volumetric approaches consider the action as a single 3D volume, which is generated by concatenating 2D images along time axis. Subsequently features are extracted from this volume to form a single representation for the whole action video. Volumetric approaches are further classified into 3 subcategories based on the representation they use for the action: Space-time shapes use 2D or 3D shapes to model the action. Trajectory-based methods use a set of 2D or 3D points or trajectories for modeling human actions. Finally histogram-based methods use distribution of features or visual words to represent human actions.

2.2.2.1 Space-time Shapes

Space-time shape methods usually extract silhouettes from each frame and either concatenate them along time or superimpose them on the same image. An example for the 2D shape is the temporal templates of Bobick and Davis [2]. They aggregate all the extracted silhouettes pertaining to one action. There are two methods of aggregation. First they use equal weights for all frames, and call the resulting representation the Motion Energy Image (MEI). In the other method they use decaying weights for frames, so that newer frames have higher weights and older frames have lower weights. This results in Motion History Image (MHI) (Fig. 2.3). From these template images, scale invariant Hu moments are extracted to be compared. Although this method performs well for simple actions like bending, sitting down and crouching, for more complicated actions, the discriminative power is reduced because of overwriting in the MEI and MHI. Also,
extracting the silhouettes may be complicated in cases of non-uniform backgrounds or moving cameras. Researchers have also tried to model the video as a 3D spatio-temporal object. Blank et al. [3] used background subtracted silhouettes and pile them together to make a 3D space-time volume (Fig. 2.4). The Poisson’s equation is solved on this 3D model to extract the 3D shape descriptors. In this method the quality of extracted silhouettes influences the results. This method has achieved the state of the art results on the foregrounds extracted from Weizmann dataset. Rodriguez et al. [13] have generalized the maximum average correlation height (MACH) filters which have been used on the images for 3D space-time volumes. For each action class a synthesized filter is generated. Action recognition is done by computing the response of the filters on the query action.

In general space-time shapes rely on silhouettes which are informative cues for action recognition. They are able to capture the spatio-temporal characteristics of motion with higher computational efficiency. Moreover no explicit body model is required. On the other hand, extracting silhouettes is not always straightforward especially when the background is not simple or in cases of severe jitter.

2.2.2.2 Trajectory-based methods

Trajectory-based approaches model actions as a set of space-time trajectories. In this representation, a subject is modeled as a set of joint positions. As the subject execute an action the position of these joints change forming 3D XYT or 4D XYZT trajectories. The work of Johansson [52] suggested that the trajectories of important joints are sufficient to recognize human actions. Using the 4D XYZT space, Sheikh et al. [53] modeled
actions as a set of 13 joint trajectories. Subsequently, affine projection is used to obtain XYZ trajectories which are view-invariant. The similarity between trajectories of two sequences is used for recognition. Yilmaz and Shah [54] also used a set of 4D joint trajectories. They proposed a method for comparing the actions from moving cameras. Assigning regions of video to the relevant action is addressed by Raptis et al. [55]. This segmentation is done by a graphical model which uses appearance and motion constraints for individual parts as well as pairwise dependencies among them. They have obtained the state of the art recognition accuracy on UCF dataset.

Point trajectories are able to capture the spatio-temporal information efficiently. But they require accurate tracking methods which may fail in the cases of high-speed targets, occlusions or cluttered backgrounds.

2.2.2.3 Histogram-based methods

Histogram-based methods typically extract spatio-temporal interest points from the space-time volume of action. The extracted features are usually used in a bag-of-words (BOW) framework to recognize the actions. The BOW framework involves clustering of the features for constructing a codebook of visual words and quantizing the features of the query based on the obtained codebook. Subsequently a histogram is formed based on the query and used for the classification.

Different methods have used different spatio-temporal features. Schuldt et al. [4] have used a generalization of famous Harris corner detector to the spatio-temporal case to define a response function. Positions of features are defined as the local maxima of this response function. An example of these detected features is shown in figure 2.5. In these feature positions, jets are extracted which are simply spatial or temporal gradients. Dollar et al. [35] argue that simple extension of 2D interest points to the spatio-temporal domain is not adequate and hence, they have used a Gabor filter in the time domain and a Gaussian kernel in the spatial domain. Cuboids are extracted in the neighborhood of detected interest points. The flattened gradients inside the cuboids followed by PCA are used as descriptors. They have used both nearest neighbor and SVM as their classifier. Niebles et al. [56] has followed the same manner, but instead they have used Probabilistic Latent Semantic Analysis (pLSA) for classification. Also there are approaches which use
a large bank of filters. Jhuang et al. [57] in an inspiration from human cortex system has used a bank of Gabor filters in different directions. They have just used the motion information. As a follow-up to this work, Schindler and Van gool [58] has used the shape information in addition to motion data. Zelnik-Manor and Irani [59] use histograms of normalized gradients of spatio-temporal volumes at different temporal scales.

While most of the methods based on spatio-temporal features rely on discriminative local descriptors, Bregonzio et al [60] have used the global distribution of features to discriminate between them. This is done by extracting holistic features from cloud of interest points which are aggregated over different temporal scales. While majority of the features are computed based on image intensity, Rapantzikos et al. [61] have proposed a saliency-based feature which have combined color and motion apart from intensity.

High level semantic concepts denoted as attributes have been used by Liu et al. [62]. The attributes are computed from the training data using an information theoretic approach. A latent SVM framework is used where the latent variables indicate the importance of each attribute for each action. They show that their approach is capable of classifying novel actions which no training data is available.

While it is prohibitive to label huge dataset of video clips, like YouTube for example, it is much easier to label a small dataset and use a mechanism to leverage these information in order to automatically annotate the videos. Motivated by this idea, Wang et al. [63] have integrated semi-supervised learning and shared structure analysis in order to perform action recognition.
Instead of first tracking humans and then performing action recognition, Khamis et al. [64] construct a network flow-based model that links the detected bounding boxes throughout the video sequence while recognizing the human action. A constrained minimum cost flow problem is solved in order to perform action recognition which leverage both appearance similarity and action transition links.

Li and Zickler [65] in an attempt to do cross-view action recognition, use a sequence of linear transformations as a virtual view which connects the source view to the target view.

While most of the researchers focus on recognition of actions from normal RGB cameras, Wang et al. [66] propose a method for recognizing actions from a depth sensor. Fourier temporal pyramid is introduced as a suitable feature for depth data which is capable of both human motion and human-object interactions. An actionlet ensemble model is learned for representation of actions, where an actionlet being a conjunctive structure defined on the base features.

The state of the art result on KTH dataset has been obtained by Lin et al. [67]. They have modeled human actions as prototype trees in the joint space of shape and motion. The learning of the trees is done by hierarchical k-means clustering. Action recognition is done by dynamic prototype sequence matching via maximizing the joint likelihood.

Histogram-based methods mostly rely on spatio-temporal features which unlike silhouettes or point trajectories are fast and easy to extract. However by using orderless local interest points we may lose the global structural information which might be helpful in classification. Therefore, Thi et al. [68] have considered global configuration of local features. Specifically they propose using two alternatives for structural learning of human actions: Dynamic Conditional Random Field and Structural Support Vector Machine. They have also presented a novel feature selection process denoted as Sparse Hierarchical Bayes Filter.

2.3 Conclusions

In this chapter we reviewed different features used for recognizing actions in videos as well as various approaches which address this problem. Common features which are
exploited are optical flow, point trajectories, silhouettes and spatio-temporal interest points. Optical flow infers both the moving areas as well as their velocity, but it is not robust to noise and illumination changes. Point trajectories require accurate tracking methods which fail with fast motion, occlusions and cluttered backgrounds. Silhouettes are informative features that infer the shape but requires foreground extraction which is difficult in complex backgrounds. On the other hand, extracting spatio-temporal interest points are easy and reliable; however they limit the description to local regions and do not present a global view of action.

We reviewed different approaches towards human action recognition and classified them into two broad categories of sequential and volumetric approaches. Sequential approaches represent action as a sequence of individual observations while volumetric approaches treat the entire action as a single volume. Sequential approaches are further categorized into exemplar-based and state-space approaches. Exemplar-based methods describe the action categories based on the training samples while state-space approaches construct a statistical model for each category. Exemplar-based approaches are used for simpler actions while state-space methods are applied when actions are more complex. Volumetric approaches that consider an action as a volume are clustered into three sub-categories of space-time shapes, trajectories and histograms based on the representation day use for modeling an action. Space-time shapes commonly use silhouettes, while trajectories and histograms exploit trajectories of important joints and spatio-temporal interest points respectively.
Chapter 3

Human Action Recognition using Pose-based Discriminant Embedding

In this chapter we propose a novel embedding for learning and recognizing human actions from a sequence of silhouettes. In the embedded space, similar poses from intra-class actions are embedded close together while other poses are as far apart as possible. This ensures that the sequences are discriminative in terms of the distance between them. Poses are chosen equidistantly from one action period. The action period is computed by a modified correlation-based method. Action recognition is achieved by comparing the projected sequences in the low-dimensional space using distance measures between sequences in a nearest neighbor framework. Several experiments are carried out on three popular and challenging databases. The method is shown not only to classify the actions effectively, but also to be robust to additive noise and tolerance to occlusion, deformation and change in view point. The method outperforms other classical dimension reduction techniques and performs faster by choosing less number of postures.

3.1 Introduction

In chapter 2 we review features used in action recognition. Some of the common ones are optical flow [24], point trajectories [69] and spatio-temporal interest points [4, 35, 56, 70]. Optical flow vectors are often inaccurate when the videos are of low quality and especially when motion is not smooth. Moreover, it is not robust to illumination changes. Likewise, point trajectories need accurate tracking methods which could fail in
cases of fast moving subjects, occlusions and cluttered backgrounds. Also by using sparse representation of interest points, we lose the global structural information, which could otherwise help in the recognition process. On the other hand, silhouettes are informative features for describing actions [3, 5, 28–30, 41, 71]. They are able to capture the spatio-temporal characteristics of motion with possibly lower computational costs [71]. There is no need for an explicit model of the human body. Furthermore, recent advances in foreground extraction from complex backgrounds and in the presence of camera motion have benefited from improved models for segmentation and global motion extraction. In particular, segmentation algorithms have benefited from the area of alpha matting in which a pixel is composed of both background and foreground components and the problem is to estimate these components [72–74].

There have been two general frameworks for using silhouettes in action recognition. Some approaches classify action sequences on a frame-by-frame basis [30]. Thus each frame is independently classified as belonging to one of the actions. The label for the query sequence is obtained based on a voting scheme. All these approaches belong to the frame recognition framework. These methods ignore the temporal information and kinematics which is useful in classification. Then, there are methods that classify the sequence as a whole [28]. These approaches belong to the sequence recognition framework. These methods compare sequences based on distances like Spatio-temporal Correlation Distance (SCD) or Hausdorff distance, which are defined between sequences of points. Human action, when represented as a sequence of silhouettes, can be considered as a function of time in which the silhouette of the body changes gradually. Thus, motion information is also included in this kind of representation without using expensive motion features that are difficult to extract. In this work we utilize the latter framework.

Silhouettes can be considered as points in high-dimensional image space. Consequently, action sequences are described as data trajectories inside image space. Recognition methods which operate in this high-dimensional space suffer from the curse of dimensionality. In addition, the information provided in the high-dimensional image space is way more than required to describe an action. Moreover, the structure of the human body imposes a constraint on possible postures. Hence, it is more efficient to analyze action trajectories in a lower dimensional space. We call this subspace the action
3.1.1 Motivation and Overview of Approach

There have been previous efforts at learning an efficient action space in order to classify actions. Accordingly, general dimension reduction techniques such as Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA), Locality Preserving Projections (LPP) [28], Locally Linear Embedding (LLE) [75], Laplacian Eigenmaps (LE) [76] and Kernel PCA [29], as well as action-specific embeddings like Local Spatio-Temporal Discriminant Embedding (LSTDE) [30] have been used. These methods are explained in more details in the next section. In all these methods, the embedding is defined based on the distance between data points rather than the distance between sequences; thus, they may be efficient in the frame recognition framework, but they are not guaranteed to give optimum results when sequences are classified. As stated earlier, in the sequence recognition framework, the query sequence is compared to the learned sequences based on distances generally defined between sets of data points, like SCD. In this work we
develop a novel embedding which is optimum in the sequence recognition framework. Specifically, the proposed embedding minimizes sum of the distances between intra-class sequences while maximizing the total distances between inter-class points.

SCD is an effective distance between ordered sequences. In order to compute SCD between two sequences, they need to have the same lengths. First, two sequences are shifted circularly to ensure that the frames corresponding to similar poses are aligned together. Then the distance between corresponding frames are calculated and summed to compute the overall distance. In order to find the optimum embedding based on SCD, we represent sequences by a fixed number of postures chosen equidistantly in one action period and obtained by interpolation. The proposed embedding is such that in the action space, same postures from same actions (executed by different individuals) are embedded close, while postures from different actions are embedded as far apart as possible. Since this embedding acts on each pose of the action, we call it Pose-based Discriminant Embedding (PDE).

Figure 3.2 shows the flowchart for the training phase of the proposed method. First the action period in the training sequences are estimated based on a modification of the method of Cutler et al. [77]. For computational efficiency only one period of each sequence is used. Sequences are then warped to have the same length. Subsequently, for intra-class sequences, similar postures are aligned together. Eventually the PDE projection matrix is computed. Details of each part are explained in the next sections.

The block diagram for the testing phase is illustrated in figure 3.3. During test, period
estimation and warping are needed only when SCD is used as the distance metric and hence, they are shown in dotted blocks. They are not required when using the Hausdorff distance, although the accuracy of recognition is increased when they are used. The query sequence is embedded into the action space based on the embedding matrix computed in the training phase and compared to the trained sequences in the low-dimensional action space using SCD or Hausdorff distance. A simple nearest neighbor classifier is used to classify the sequence. Thus in the action space the query sequence is classified as the class of the nearest trained sequence.

Our proposed method guarantees to give the most discriminant action space in order to be used with SCD. Since the sequence-based recognition framework is more efficient than frame-based, the proposed method outperforms other dimension reduction methods like PCA, LDA and LPP in action recognition. Also our method performs faster and is more computationally efficient compared to the aforementioned embeddings by using less number of frames (postures). Moreover, our method is robust to additive noise, occlusion and deformation. Also viewpoint change is tolerated up to a considerable extent.

3.1.2 Organization

This chapter is organized as follows. Section 3.2 reviews some related works on dimension reduction techniques used for action recognition. The distance metrics used between sequences are described in section 3.3. Subsequently the proposed embedding method as well as the objective functions used for optimization is detailed. Section 3.4 describes the steps required for preprocessing. Experimental results are presented and discussed in section 3.5. Finally we conclude the chapter in section 3.6.

3.2 Related Work

In this section, we review the dimension reduction methods which have been used in action recognition. While many dimension reduction techniques have been used successfully for subspace learning for face recognition, only few dimension reduction methods have been used for learning the action space for action recognition. Three of the common dimension reduction techniques including PCA, LDA and LPP have been used by Wang
Chapter 3. Human Action Recognition using Pose-based Discriminant Embedding

and Suter [28] to discover the underlying action manifold. PCA simply chooses the directions in which the data has maximum variance [78]. It is intrinsically an unsupervised technique. LDA is a supervised method, which tries to make data discriminative for better classification [79]. This means that in the embedded space, intra-class data points are as close as possible while inter-class points are as far apart as possible. Although it is generally believed that LDA performs better than PCA, yet when the training set is small, PCA can outperform LDA [80]. LPP tries to preserve the distance between data points in the target space so that local structure is preserved [81]. It is used in either supervised or unsupervised manner.

All the methods mentioned so far are linear. LE [76], LLE [75] and kernel PCA [29] are nonlinear techniques which have been used for action recognition. Experiments performed by Wang and Suter [28] have verified that linear methods (PCA, LDA and LPP) outperform the nonlinear ones (LE and LLE) in action recognition. The reason for this is the complexity in parameter adjustment and extrapolation in these nonlinear methods [28]. Another disadvantage of nonlinear dimension reduction techniques is that they are computationally expensive.

The concept of Canonical Correlation Analysis (CCA) or Principal Angles [82] have also been used for manifold learning which models the set of intra-class frames as a hyper plane. Similar to optimization concept in LDA, Kim et al. [83] have developed a discriminant analysis using CCA. In this framework which is used to classify the image sets, the canonical correlations of inter-class sets are maximized, while minimizing the canonical correlations related to intra-class sets. The optimal embedding is computed in an iterative learning manner. In a similar work, Wu et al. [41] have proposed an online framework, which incrementally update the discriminative model upon adding online training samples using the eigenspace merging algorithm. These methods ignore the temporal constraints in action sequences which may be useful in classification. Jia and Yeung [30] have developed another embedding which is discriminative in two folds: spatial and temporal. Their criteria for spatial discrimination are similar to LDA. However, for temporal discrimination, they find the embedding such that the principal angles between inter-class temporal subspaces are maximized. The temporal subspace is formed by a short video segment around each frame. Their method does not create a dichotomy
between spatial and temporal analysis since the action is represented as a sequence of silhouettes that contain spatio-temporal changes.

As an instance for considering temporal constraints, Nayak et. al in [84] have proposed learning a subspace of distributions for recognition of articulated activities. The action is represented as frame-wise distributions of low level features such as orientation, color, or relational distributions. As time grows, the configuration of parts change which result in changing the distribution in the latent space. So the whole action will result in a trajectory, which can be compared in the latent space. The experiments have been done on gesture recognition and classification of human-human interaction sequences.

While most of the corpus of manifold learning approaches are based on representing images or frames as pixel-wise vectors, Torki and Elgammal [85] have proposed to learn manifolds from a collection of local features such that it captures the feature similarities and spatial structures. By choosing proper affinity metrics between feature descriptors and spatial coordinates, first the training set is embedded into a new space. Subsequently features of the new image is embedded using a coordinate propagation method. This method has been successfully verified on shape and object recognition datasets.

3.3 Pose-based Discriminant Embedding

In the sequence recognition framework, sequences of silhouettes are embedded into a lower dimensional space called action space. The sequences are compared in the action space using distances defined between sets of data points.

Each frame with the resolution of $M \times N$ is converted into a vector $f$ of dimension $h = M \times N$ in the lexicographic order. Let $f_i(t)$ represent the frame at the time $t$ from the $i^{th}$ sequence. Then the $i^{th}$ sequence, $F_i$ can be considered as a function of time by $F_i = f_i(t), t = 1, \ldots, N_i$, where $N_i$ is the number of frames in $F_i$. The total number of training samples for $n$ training sequences is $N_t = N_1 + \ldots + N_n$. The overall training set is denoted by

$$X = [f_1(1), f_1(2), \ldots, f_1(N_1), f_2(1), \ldots, f_n(N_n)]$$
$$= [x_1, x_2, \ldots, x_{N_t}]$$ (3.1)
Thus, $X$ is a matrix of size $h \times N_t$. Each data point $x_i$ in the $h$-dimensional image space is embedded into a point $y_i$ in the $l$-dimensional ($l << h$) action space by $y_i = A^T x_i$, where $A$ is the embedding matrix. Specifically, the sequence $F_i = f_i(t), t = 1, \ldots, N_i$ is embedded into sequence $Q_i = q_i(t), t = 1, \ldots, N_i$ in the action space. Embedded sequences are compared based on the distances defined between sets of points. We first review these distances and then describe how the proposed embedding is computed.

### 3.3.1 Distances between sets of points

#### 3.3.1.1 Median Hausdorff Distance

The Hausdorff distance measures the similarity of two data point sets, by finding the points in one set which are close to points in the other set and vice versa. The MHD from a sequence $Q_1$ to a sequence $Q_2$ is defined as [28]:

$$d_{\text{MHD}}(Q_1, Q_2) = \text{median}_i(\min_j(\|q_1(i) - q_2(j)\|)),$$

(3.2)

where $i$ and $j$ refer to time instances. We believe that \textit{Median Hausdorff Distance} is a better choice than \textit{Mean Hausdorff Distance} since the former is not affected by outliers. To incorporate symmetry, the final distance measure used is

$$D_{\text{MHD}}(Q_1, Q_2) = d_{\text{MHD}}(Q_1, Q_2) + d_{\text{MHD}}(Q_2, Q_1)$$

(3.3)

When using MHD, two sequences do not need to be of the same length. Moreover, MHD ignores the order of frames in the sequences.

#### 3.3.1.2 Spatiotemporal Correlation Distance

SCD considers the arrangement of frames in sequences; thus, it is more efficient for sequence recognition and particularly action recognition. The two sequences should have the same length in order to be compared. SCD between sequences $Q_1$ and $Q_2$ is defined as [28]:

$$D_{\text{SCD}}(Q_1, Q_2) = \min_b \sum_{t=1}^{T} \|q_1'(t) - q_2'(t + b)\|^2,$$

(3.4)

where $Q'_1 = q'_1(t)$ and $Q'_2 = q'_2(t); t = 1, \ldots, T$, are warped versions of the sequences $Q_1$ and $Q_2$. In section 3.4, we explain how to warp the sequences into the same length.
variable \( b \) stands for circular time shifting to align the corresponding frames together for comparison. By aligning we ensure that equation (3.4) gives the minimum distance among the different alignments. SCD is basically the sum of Euclidean distances between corresponding points that represent frames in the embedded space.

As explained earlier, MHD ignores the order of points in the sequences. However, the order of silhouettes in the sequences can be useful in the classification process. Thus in order to find the optimal embedding we use SCD, which is based on ordered sequences. Note that SCD needs the action period to be computed. Since estimating the action period is not always easy in real videos, in this work we also use MHD in order to compare the sequences.

### 3.3.2 Optimal Embedding Computation

In this work we propose an embedding such that in the embedded space (action space), based on SCD as the distance metric, the intra-class sequences are as close as possible while the inter-class sequences are as far apart as possible. We represent both these criteria in terms of two novel affinity matrices. Each of these criteria is represented by novel objective function which is finally combined into an overall optimization problem. We follow the general graph embedding framework [30, 86] to solve the overall optimization problem. Let \( A \) denote the embedding matrix. In order for intra-class sequences to be as close as possible in the action space, the sum of all pairwise SCD between embedded intra-class sequences should be minimized with respect to \( A \):

\[
\min_A \sum_c \sum_{i,j \in C_c} D_{SCD}(A^T f_i, A^T f_j) = \min_A \sum_c \sum_{i,j \in C_c} \sum_{t=1}^T \| A^T f_i(t) - A^T f_j(t) \|^2; \quad (3.5)
\]

where \( C_c \) represents the set of all sequences belonging to class \( c \). During preprocesssing, the sequences are warped to the same length \( T \) and the intra-class sequences are aligned by applying the circular time shifting in equation (3.4). In order to rewrite equation (3.5) in terms of \( x \), we define the \( N_t \times N_t \) matrix \( W_p \) such that

\[
W_{p_{ij}} = \begin{cases} 
1, & \text{if } x_i \text{ and } x_j \text{ are similar postures from the same action.} \\
0, & \text{otherwise.} 
\end{cases} \quad (3.6)
\]
This way the data points \( \{x_1, x_2, \ldots, x_N\} \) are modeled as the nodes of a graph \( G_p \) with \( W_p \) as its affinity matrix. Therefore equation (3.5) is converted to

\[
\min_A \sum_{ij} \|A^T x_i - A^T x_j\|^2 W_{pi,j}. \tag{3.7}
\]

For ease of derivation, we rewrite the objective function of (3.7) in the trace format as

\[
\frac{1}{2} \sum_{ij} \|A^T x_i - A^T x_j\|^2 W_{pi,j} = \frac{1}{2} \sum_{ij} Tr\{A^T(x_i - x_j)(x_i - x_j)^T A\} W_{pi,j} = \frac{1}{2} Tr\{A^T \sum_{ij} ((x_i - x_j) W_{pi,j}(x_i - x_j)^T) A\} = Tr\{A^T (XD_p X^T - XW_p X^T) A\} = Tr\{A^T XL_p X^T A\}, \tag{3.8}
\]

where \( X = [x_1, x_2, \ldots, x_N]\), \( D_p \) is a diagonal matrix with column (or row) sums of symmetric \( W_p \) as entries, and \( L_p = D_p - W_p \) forms the Laplacian matrix of the graph \( G_p \).

Since each entry of \( D_p \) is a row (or column) sum of \( W_p \), a large value of \( D_p(i, i) \) indicates the higher importance of the associated point. Therefore, the following constraint is imposed:

\[
Tr\{A^T XD_p X^T A\} = 1, \tag{3.9}
\]

so that the objective function in equation (3.8) turns into

\[
\min_A \{1 - Tr\{A^T XW_p X^T A\}\}, \tag{3.10}
\]

or

\[
\max_A Tr\{A^T XW_p X^T A\}. \tag{3.11}
\]

In order for inter-class sequences to be as far apart as possible based on SCD, we need to align each pair of inter-class sequences when compared together. But sequences are already aligned based on intra-class labels. So the alignment cannot be done for inter-class labels. For instance, suppose we have sequences \( F_1, F_2, F_3 \) and \( F_4 \) for which \( F_1 \) and \( F_2 \) belong to class \( C_1 \) and \( F_3 \) and \( F_4 \) belong to class \( C_2 \). \( F_2 \) is aligned based on \( F_1 \) and similarly \( F_4 \) is aligned based on \( F_3 \). When comparing \( F_1 \) and \( F_4 \), \( F_4 \) cannot be aligned based on \( F_1 \), because it has already been aligned based on \( F_3 \). Thus for optimization of
inter-class sequences, we consider the distance between data points rather than sequences as for the previous discriminative embeddings. So we define the affinity matrix $W_c$ such that

$$W_{c_{ij}} = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ belong to different actions.} \\ 0, & \text{otherwise.} \end{cases}$$

Consequently the objective function which has to be maximized is

$$\max_A \sum_{ij} \| A^T x_i - A^T x_j \|^2 W_{c_{ij}}. \quad (3.13)$$

Similar to equation 3.8, the objective function of (3.13) can be written in trace form as

$$\max_A \text{Tr} \{ A^T X L_c X^T A \}, \quad (3.14)$$

where $L_c = D_c - W_c$ ($D_c$ is the diagonal matrix with column (or row) sums of $W_c$ as entries). Given the objective functions in equations (3.14) and (3.11), together with the constraint in equation (3.9), we have the overall optimization problem for PDE as:

$$\max_A \{ \text{Tr} \{ A^T X W_p X^T A \} + \text{Tr} \{ A^T X L_c X^T A \} \} \text{ s.t. } \text{Tr} \{ A^T X D_p X^T A \} = 1. \quad (3.15)$$

Equation (3.15) is equivalent to the following trace ratio optimization problem:

$$\max_A \frac{\text{Tr} \{ A^T X (W_p + L_c) X^T A \}}{\text{Tr} \{ A^T X D_p X^T A \}}. \quad (3.16)$$

Since this problem does not have a closed-form solution [87], it is usually simplified into the ratio trace problem of $\max_A \text{Tr} \{ (X D_p X^T)^{-1} (X (W_p + L_c) X^T) \}$ which can be solved using the generalized eigenvalue problem [87]. An optimal $A$ can be obtained by finding the $l$ largest eigenvalues of the following generalized eigenvalue problem:

$$(X (W_p + L_c) X^T) a = \lambda (X D_p X^T) a. \quad (3.17)$$

The generalized eigenvalue problem is solved using the spatially smooth subspace learning algorithm proposed by Cai et al. [88]. Their regularized method explicitly considers the spatial relationship between entries of a matrix which results in smoother subspaces. They optimize a Laplacian penalty function which constrains the coefficients to be spatially smooth. The proposed embedding method projects sequences into a space in which
the intra-class sequences are close together (in terms of SCD) and also the inter-class frames are as far apart as possible. While the latter is similar to LDA, the former characteristic enables our method to clearly outperform LDA and other similar discriminant embeddings in the sequence recognition framework. LDA tries to minimize the distance between intra-class points, but may not necessarily give the least SCD possible between intra-class sequences, which our method seeks. When comparing the sequences based on SCD it will have less classification error compared to LDA. This superiority becomes even more obvious in the cases of similar inter-class actions like run, walk and skip as well as when the silhouettes are noisy. In these situations inter-class sequences may be originally closer than intra-class ones. PDE enforces the intra-class sequences to be closer in the action space compared to LDA leading to less error in classification.

The other characteristic of the proposed embedding is that in the action space, similar postures from the same actions (performed by different subjects) are embedded as close as possible. Figure 3.4 shows two trajectories of run performed by two different persons. The change in appearance has resulted in deviation of postures in the action space. The embedding aims to remove these noise-like deviations from trajectories and make them as close as possible so as to minimize the classification error. In other words, it aims to make all intra-class trajectories performed by different subjects, coincide in the action space.
3.4 Preprocessing

As shown in figure 3.2 we perform some preprocessing on the training sequences before computing the PDE. Preprocessing involves period estimation, warping and aligning. The first two steps reduce the computational complexity as well as time by choosing less number of frames. Similar postures from intra-class actions are also aligned for embedding computation. Moreover, as illustrated in figure 3.3 the query sequences are also preprocessed for period estimation and warping.

3.4.1 Period Estimation

Action sequences can be considered as periodic repetitions of an action cycle. Using a single period is much more computationally efficient than using the entire length of the video sequence. For computing the action period, we use the method of Cutler and Davis [77], which is also used in [28]. However, we modify it slightly for more accurate estimation. The distance between frames at times $t_1$ and $t_2$ are computed as [77]:

$$S_{t_1,t_2} = \sum_{(x,y)} |B_{t_2}(x,y) - B_{t_1}(x,y)|,$$  \hspace{1cm} (3.18)

where $B_{t_1}$ and $B_{t_2}$ are the silhouettes in times $t_1$ and $t_2$ respectively which have been centered and normalized into the same dimensions. $S$ will have a periodic pattern as shown in figure 3.5(a), in which darker regions indicate less distance. For periodic action sequences (like walk in Weizmann database), dark lines are arranged parallel to the diagonal of $S$. To determine the action period, one arbitrary column vector $z$ of $S$ is chosen and linearly detrended to obtain the new vector $\hat{z}$ (Fig 3.5(b)). Then the autocorrelation of $\hat{z}$ is computed (figure 3.5(c)). In [28] the action period is estimated as the mean distance between each pair of consecutive peaks in the aforementioned autocorrelation function. Here we use median instead of mean since it leads to more stable results.

We do not use the zero-derivative method to find peak positions as in [28] since a little noise will result in many false detections. Instead, we pick a point as the peak whose left and right neighbors have lower values by some margin [89]. In figure 3.6 the false peak detections which result from applying the zero-derivative method is illustrated, which
Figure 3.5: Period estimation for walk sequence pertaining to person 1 (Daria) of Weizmann database \cite{3} with a length of 84, (a) similarity $S$, (b) $\hat{z}$, the middle column vector $z$ of $S$ which is linearly detrended, (c) its autocorrelation.

Figure 3.6: False peak detections by zero-derivative method indicated by red vertical lines.
3.4.2 Warping

In order to use SCD, the sequences should have the same length. So we need to warp all the training sequences into the same global length before computing PDE. Warping the test sequence is also necessary when using SCD as the distance metric. However, it is not required for MHD due to the point-set matching properties of this measure.

By warping, we choose less number of frames to be processed. Thus we decrease the computational complexity and time of the processing with the accuracy almost remaining the same. Warping is done on one action period. The action cycle is warped to the length $T$, by selecting $T$ time instances equidistantly from the period, starting from the beginning, and interpolating frames at the selected times. The interpolation is done using the existing frames by bicubic interpolation technique. Figure 3.7 illustrates the warping procedure. The green circle markers show the time instances for existing frames. If the action cycle is $P$, the posture in time instance $P + 1$ is similar to the posture in time 1. The blue triangle markers stand for the time instances of interpolated frames.
Figure 3.8: Aligning two warped sequences of action run: The first pose of sequence 1, is similar to the 5th pose of sequence 2. The other poses are aligned in a cyclic manner. For this example: $b = 4$.

time samples are denoted by $t_1, t_2, \ldots, t_T, t_{T+1}$. They are chosen equidistantly during one period. So we have

$$\frac{t_i - t_1}{t_{T+1} - t_1} = \frac{i - 1}{(T + 1) - 1}; i = 1, 2, \ldots, T. \quad (3.19)$$

Since $t_1 = 1$ and $t_{T+1} = P + 1$, the time instances are chosen by

$$t_i = 1 + \frac{P}{T}(i - 1); i = 1, 2, \ldots, T. \quad (3.20)$$

$T$ is chosen less than the minimum possible action period. The effect of $T$ is studied in section 3.5.

3.4.3 Aligning

After warping, we need to align the similar poses of training sequences in the same order before computing the embedding matrix. For this purpose, we employ equation (3.4) used for computing SCD. Here the objective is to compute $b$, the circular shift and not the distance. For each action in the training set, one sequence is considered as the reference for aligning (e.g. sequence pertaining to person 1). The rest of the intra-class sequences are aligned with the reference sequence. For example figure 3.8 shows two warped sequences of action run. After aligning, the first pose of the reference sequence is similar to the 5th pose of sequence 2 resulting in $b=4$. 

38
3.5 Experimental Results

We have performed several experiments to show the efficiency and effectiveness of PDE for action recognition. We have used three common action data sets: Weizmann database [3], Maryland database [5] and KTH database [4] to measure the goodness of PDE for dealing with intra-class variance and inter-class similarity among actions. Weizmann robustness dataset is used to test occlusion and viewpoint change. The execution rate variance is tested by Maryland dataset. In order to have a more comprehensive study we have also experimented on KTH dataset which is more challenging than the first two.

Segmenting the foreground is not the main interest in our work, so we use the silhouette masks which are available in Weizmann and Maryland datasets. For KTH dataset we manually extract silhouettes as described in subsection 3.5.3. All silhouette frames are centered and normalized into the same dimension which is 64 × 48 for Weizmann and Maryland datasets and 80 × 48 for KTH dataset.

To avoid the singular matrix problem in the optimization of equation (3.15), we preprocess the data using PCA so that 98% information is kept in the sense of low rank approximation.

3.5.1 Experiments on Weizmann Database

This database contains ten action classes performed by nine different human subjects. The actions include bending (bend), jumping jack (jack), jumping-forward-on-two-legs (jump), jumping-in-place-on-two-legs (pjump), running (run), galloping sideways (side), skipping (skip), walking (walk), waving-one-hand (wave1), and waving-two-hands (wave2) [3]. In all the experiments, we use the leave-one-out cross validation method, i.e., each time we leave one sequence out for testing and train with the remaining sequences in the dataset. We report the average of recognition results.

3.5.1.1 Results and Analysis

We warp both the training and test sequences into the temporal duration $T$. Since the minimum length of the periods is 9 (run), we let $T$ to vary from 3 to 9 in order to study the effect of $T$. For each value of $T$, we change the dimension ($l$) from 1 to 200. SCD is used
as the distance metric. The results are shown in figure 3.9. We achieve recognition rates close to 100%, which clearly shows the efficiency of our method in learning the action space. Recognition rate of 100% is achieved for $T = 8$ and $T = 9$. As seen from the plots in figure 3.9, the accuracy usually does not change considerably with small changes of $l$. The sudden drop off for $T=3$ and $T=4$ is likely due to *curse of dimensionality*. In other words, by increasing the dimension from this point forward the number of training samples are no longer sufficient to learn the subspace. This occurs for small $T$s, since the number of training data points are much lower than for bigger $T$s, given the training sequences. The mean training time on a 2.67 GHz CPU ranges from 0.57 sec for $T = 3$ to 3.09 sec for $T = 9$, which is considerably low. Also the testing lasts for less than 0.01 sec which is significantly fast.

To study the effect of $T$, we compare the maximum and also mean of recognition accuracy in the stable interval from the plots in figure 3.9 for different values of $T$. The
Figure 3.10: Studying the effect of $T$: comparing the maximum and mean of recognition rate in the stable interval from the plots in figure 3.9 for different values of $T$.

comparison is illustrated in figure 3.10. As seen from the figure, the accuracy is not so sensitive to the change of $T$. The best results (100%) occur for $T = 8$ and 9. By reducing $T$, the accuracy almost decreases, since smaller $T$s have less number of postures to discriminate between actions.

We also explore using MHD as the distance measure. Similar to figure 3.9 the results using MHD when both the training and test sequences are warped is shown in figure 3.11. Here, too, we achieve recognition rates close to 100%. We obtain 100% recognition accuracy for $T = 6$. Similar to SCD, there are drop offs for $T = 3$ and $T = 4$. The test time using MHD ranges from 0.02 sec for $T = 3$ to 0.07 sec for $T = 9$. Testing takes longer compared to using SCD. Since the distance is not used during training phase, the time for training is the same as SCD. Comparing figures 3.9 and 3.11, the results using MHD seems smoother because SCD is a sum of $T$ distances while Hausdorff distance is simply one of the distance values (the median value).

In order to examine the effect of $T$, comparison between maximum and average accuracy of different $T$s is shown in figure 3.12. For MHD, $T=6$ which is the middle value in our range (3-9) has the best results. Similar to SCD, the accuracy for small $T$s are
lower. Moreover, for MHD the accuracy for large $T$ values are slightly lower than $T = 6$. This is probably because adjacent frames in a sequence will be similar and they will be no longer discriminant.

When using MHD, query sequences are not required to have the same length. So we also investigate using the test sequences without warping. From figure 3.13, we have accuracy near 100%, which shows the efficiency of our method even when the query sequence is not warped. The maximum recognition rate without warping the test sequence, has dropped slightly to 98.89%. This is due to the lower distances between warped intra-class sequences. We expect the minimum distance between warped intra-class point sets in equation 3.2 to be the distance between similar postures which is small. However, this is not necessarily true without warping.
Figure 3.12: Studying the effect of $T$: comparing the maximum and mean of recognition rate in the stable interval from the plots in figure 3.11 for different values of $T$.

Figure 3.13: Recognition accuracy versus dimension, using the test sequences without warping for $T = 6$ and MHD as distance.
Table 3.1: Comparison of different dimension reduction methods. The number in parenthesis shows the optimum dimension which is found empirically. $T = 6$ is used for PDE. The experiment is done using a single action cycle.

<table>
<thead>
<tr>
<th></th>
<th>PCA</th>
<th>LDA</th>
<th>SLPP</th>
<th>PDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognition Acc.</td>
<td>95.56% (16)</td>
<td>91.11% (9)</td>
<td>93.33% (8)</td>
<td>100% (156)</td>
</tr>
<tr>
<td>Mean Train Time</td>
<td>05.94 (11.7 with preprocessing)</td>
<td>41.41</td>
<td>40.39</td>
<td>01.41</td>
</tr>
<tr>
<td>(sec)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean Test Time</td>
<td>00.26</td>
<td>00.26</td>
<td>00.26</td>
<td>00.04</td>
</tr>
<tr>
<td>(sec)</td>
<td></td>
<td></td>
<td></td>
<td>(0.15 with preprocessing)</td>
</tr>
</tbody>
</table>

3.5.1.2 Comparison with different dimension reduction methods

In this section we compare PDE with well-known dimension reduction methods including PCA, LDA and supervised LPP (SLPP). In table 3.1, we show the recognition accuracy using different dimension reduction methods compared to PDE and also the average times needed for training and test. Only one period of each sequence is used in the experiments. After embedding into action space, the sequences are compared using MHD, since they have different lengths. The recognition accuracy for PDE is the most, since using PDE the distance between embedded intra-class sequences is the least while the inter-class points are as far as possible. PCA has the second highest accuracy because of the small training set of Weizmann database, since PCA performs better in small training sets [80]. SLPP outperforms LDA. This is probably because LPP is more capable in learning the nonlinear structures of action manifolds. The training time for computing PCA is lower than LDA and SLPP, since PCA is solving a simple eigenvalue problem, but LDA and SLPP involve solving a generalized eigenvalue problem with the same number of samples. Computation of PDE has the least computational time. Although PDE requires solving a generalized eigenvalue problem, but the number of samples used for PDE is much less than other methods. Note that solving a generalized eigenvalue problem is of cubic-time complexity with respect to the number of samples [90]. For instance in Weizmann dataset, if we leave the sequence of run performed by the first person aside for test and train with the remaining sequences, for $T = 6$, the number of samples (frames) for PDE is 540 while the number of samples for other methods is 2010, which is an enormous saving in time and complexity. Considering the time needed for warping (10.11 sec) and aligning (0.18
the total time needed for training our method (11.7 sec) is still lower than LDA and SLPP. Also considering the average time needed for warping each sequence (0.11 sec), our method needs the least time for query classification. This faster and more effective computation is a great advantage especially when time and memory is critical. If speed is critical, we can choose a smaller $T$ without compromising much on the accuracy.

In order to show how discriminant each method is, the visualization of data points in the action space is shown in figure 3.14, where the 3D subspace regarding the first 3 main components is illustrated. The points with the same color belong to the same action. Note that here we are studying the distribution of data points and not the sequences. From the figure 3.14, PDE appears to have better clustering effect compared to other methods. PCA is the least discriminative one since it ignores the class labels. The data points embedded by SLPP seem to be more discriminant than those projected by LDA.
Table 3.2: Comparison with recent results on Weizmann dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDE</td>
<td>100%</td>
</tr>
<tr>
<td>Wang and Suter [28], Lin et al. [67], Schindler and Van Gool [58], Fathi and Mori [91]</td>
<td>100%</td>
</tr>
<tr>
<td>Wu et al. [41]</td>
<td>98.9%</td>
</tr>
<tr>
<td>Zhong and Stevens [92]</td>
<td>98.6%</td>
</tr>
<tr>
<td>Wang and Suter [29]</td>
<td>97.8%</td>
</tr>
<tr>
<td>Zhang et al. [93]</td>
<td>92.9%</td>
</tr>
<tr>
<td>Ali et al. [25]</td>
<td>92.6%</td>
</tr>
<tr>
<td>Jia and Yeung [30]</td>
<td>90.9%</td>
</tr>
<tr>
<td>Scovanner et al. [94]</td>
<td>84.2%</td>
</tr>
<tr>
<td>Niebles and Fei-Fei [95]</td>
<td>72.8%</td>
</tr>
</tbody>
</table>

due to power of LPP to find the nonlinear structure of action manifolds.

3.5.1.3 Comparison with Recent Results on Weizmann dataset

In table 3.2, we compare PDE with some of the recent results reported on the Weizmann database. From the table, we see that our method obtains 100% recognition accuracy. Other methods with the similar accuracy are the manifold learning method of Wang and Suter [28] and those of Lin et al. [67], Schindler and Van Gool [58] and Fathi and Mori [91] that use complex features. The features used in Lin et al. [67] and Schindler and Van Gool [58] are based on information from both shape and motion, while our approach is based on simple shape features. Lin et al. [67] learn an action prototype tree in the joint shape and motion space. Schindler and Van Gool [58] extract features from shape and motion and compare them separately with learned samples. Finally the similarities are concatenated into a single vector which is classified by a bank of linear classifiers. The method of Fathi et al. [91] is based on motion features built from optical flow information and created by a variant of Adaboost. It involves the expensive and sensitive computation of optical flow. The method of Wang and Suter [28] has an accuracy equal to ours, however the processing times are significantly longer than our method. They use SLPP for learning the action space in a framework similar to ours. But they use one period more than ours, i.e. totally two action cycles for each training sequence. This way, there is more information in the training set, which possibly increases the recognition rate. So the recognition rate they achieve is more than the accuracy for
SLPP in table 3.1 i.e. 93.33%. Accordingly their training time is higher than the time in table 3.1 i.e. 40.39 sec, which is considerably longer than ours (1.54 sec). The test time (0.26 sec) is also much higher than PDE (0.02 sec). This clearly shows that our method is more efficient in finding the underlying action space.

We have compared our method with different approaches including methods using manifold learning. Methods using manifold learning and dimension reduction techniques [28–30, 41] have already been reviewed in section 3.2. Among the other methods, Zhong and Stevens [92] compute a 3D spatio-temporal volume of motion energy. Local motion descriptors are extracted from the computed volume and compared with the learned feature set in order to encode the action. Zhang et al. [93] represent action videos as motion history images and extract local features from them. Distribution of these local features over relative locations is captured by a histogram similar to shape context. Eventually each action is modeled as a 3D descriptor. Extracted point trajectories are used as features in the method of Ali et al. [25]. These features are classified based on chaotic invariants. Scovanner et al. [94] have extended the SIFT descriptors to form a 3D descriptor. In their method, videos are represented in a bag-of-words framework. Furthermore, Neibles et al. [95] use spatio-temporal features in a hierarchical framework. All these methods have recognition accuracies lower than PDE, which verifies the efficiency of our method in recognizing actions.

3.5.1.4 Robustness Test

In this section we study the robustness of our method with respect to additive noise, deformations and change in viewpoint. Since the silhouettes in the robustness test are either corrupted or deformed, the period estimation is not feasible so we use MHD which does not require estimating the action period.

**Robustness to Noise** The silhouettes that we use for the experiments are almost free of noise. To check robustness to noise, we add various amounts of synthetic noise to silhouette images to simulate corrupted silhouettes. Since the silhouette images are binary, salt and pepper noise is added. In this experiment the percentage of the affected pixels in the image is shown by noise density. We repeat this experiment for SLPP
with original silhouettes (used for PDE) as well as distance transformed silhouettes. The distance transform is used in [28] to compensate for variation of appearance among different persons. The results are shown in figure 3.15. The proposed method is clearly robust to noise, but the other methods are not. When noise is added, the distance between intra class sequences might become more than the distance between inter class sequences. The proposed method which guarantees the least distance between intra class sequences will therefore minimize the classification error. Note that noise affects the distance transformed image more than the original silhouette, because the influence of noise is greatly increased when using distance transform.

**Robustness to General Deformations** The robustness of the proposed method to some challenging factors such as rigid and non-rigid deformations, variation in clothes and motion styles and also occlusions is investigated in this section. For this purpose we use Weizmann’s robustness database for deformations which contains 10 instances of walking with general deformations [3]. Some example images and the corresponding masks are shown in figure 3.16. Each of these test sequences is compared with all the 90 actions in the Weizmann’s database to find the best match. Here we do not segment the action cycles. So for the query video, the whole sequence is used with MHD as the distance measure. The point-set matching characteristics of MHD handle the different time durations and aligning. The results are shown in table 3.3. Except for three sequences (knees up, sleepwalking and walking with a dog), all other test sequences are correctly
Figure 3.16: Sample images of Weizmann’s robustness database for deformations [3]. From left to right and from top to bottom: swinging bag, carrying briefcase, knees up, limping man, sleepwalking, occluded legs, walking in a skirt, walking with a dog, presence of a pole, normal walk, respectively.
Table 3.3: Results of Deformation robustness test.

<table>
<thead>
<tr>
<th>Test Sequences</th>
<th>Conditions</th>
<th>Classification Result</th>
<th>Best Match</th>
<th>Other Actions among the 10 Best Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>swinging bag</td>
<td>rigid deformation</td>
<td>walk</td>
<td>run, side</td>
<td></td>
</tr>
<tr>
<td>carrying briefcase</td>
<td>rigid deformation</td>
<td>walk</td>
<td>skip, run</td>
<td></td>
</tr>
<tr>
<td>knees up</td>
<td>walking style</td>
<td>side</td>
<td>pjump, walk, skip</td>
<td></td>
</tr>
<tr>
<td>limping man</td>
<td>walking style</td>
<td>walk</td>
<td>side</td>
<td></td>
</tr>
<tr>
<td>sleepwalking</td>
<td>walking style</td>
<td>skip</td>
<td>wave2, jack, side, walk, run</td>
<td></td>
</tr>
<tr>
<td>occluded legs</td>
<td>partial occlusion</td>
<td>walk</td>
<td>pjump, skip, jump</td>
<td></td>
</tr>
<tr>
<td>walking in a skirt</td>
<td>clothes</td>
<td>walk</td>
<td>side, jump</td>
<td></td>
</tr>
<tr>
<td>walking with a dog</td>
<td>non-rigid deformation</td>
<td>run</td>
<td>walk, skip</td>
<td></td>
</tr>
<tr>
<td>presence of a pole</td>
<td>occlusion</td>
<td>walk</td>
<td>side, run</td>
<td></td>
</tr>
<tr>
<td>normal walk</td>
<td>background</td>
<td>walk</td>
<td>side, run</td>
<td></td>
</tr>
</tbody>
</table>

classified as \( \text{walk} \). This is similar to [28, 29] in the number of misclassified sequences. The three misclassified sequences by our method are different with the normal walk in the sense of style and non-rigid deformation. Even for these three sequences, walk is among the 10 best matches. This shows that our method has relatively low sensitivity to changes in clothes, motion style and also rigid and non-rigid deformations and occlusion. In [28, 29] walking with a bag which is a sort of rigid deformation is being misclassified, instead of knees up.

**Robustness to Change in Viewpoint** Changing the view angle causes wide variations in motion and shape of subject and therefore result in error. In this section we study robustness of our method to variations in viewpoint. We use the Weizmann robustness dataset for viewpoints [3]. Samples images of this dataset are illustrated in figure 3.17. Here, also we use the whole sequence for the query with MHD as the distance measure. The results are shown in table 3.4. From the table, our method can tolerate up to 30 degrees change in viewpoint which is considerable. This is similar to the result of [28]. From 30 degrees to 45 degrees, the walk action is wrongly classified as \( \text{pjump} \) probably since we have less horizontal movements as well as pjump. Our method is not
Chapter 3. Human Action Recognition using Pose-based Discriminant Embedding

Figure 3.17: Sample images of Weizmann’s robustness database for viewpoint [3]. From left to right and from top to bottom: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, respectively.

Table 3.4: Results of viewpoint robustness test.

<table>
<thead>
<tr>
<th>Test Sequence (angle)</th>
<th>Classification Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best match</td>
</tr>
<tr>
<td>00</td>
<td>walk</td>
</tr>
<tr>
<td>05</td>
<td>walk</td>
</tr>
<tr>
<td>10</td>
<td>walk</td>
</tr>
<tr>
<td>15</td>
<td>walk</td>
</tr>
<tr>
<td>20</td>
<td>walk</td>
</tr>
<tr>
<td>25</td>
<td>walk</td>
</tr>
<tr>
<td>30</td>
<td>pjump</td>
</tr>
<tr>
<td>35</td>
<td>pjump</td>
</tr>
<tr>
<td>40</td>
<td>pjump</td>
</tr>
<tr>
<td>45</td>
<td>pjump</td>
</tr>
</tbody>
</table>

intrinsically designed for viewpoint change but still it manages to handle large variations in viewpoint angle which is promising.

3.5.2 Experiments on Maryland Database

This dataset [5] comprises 10 different actions performed by one person. There are 10 different instances for each action, resulting in 100 action sequences in this dataset. The instances differ on the temporal rate of execution but there are also slightly different motion styles. The actions are pick up object, jog in place, push, squat, wave, kick, bend to the side, throw, turn around and talk on cell phone. Two cameras with 45 degrees
difference in viewpoint have captured these actions. We use the videos from the frontal view. Examples of silhouettes are shown in figure 3.18. The dataset is divided into 10 sets, each containing 1 instance of all actions. Each time we leave one set out for the test and train using the remaining nine sets. The final result is the average of the ten runs. We achieved a 100% recognition rate for this database using either SCD or MHD.

The comparison between PDE and other popular dimension reduction techniques is illustrated in table 3.5. This experiment is performed using a single action cycle. From the table 3.5 our method has the best accuracy. PCA is also better than LDA and SLPP, due to small training sample. Similar to table 3.1 our method has the best classification time. Also the time needed for training is much lower than LDA and SLPP even considering the preprocessing steps.
3.5.3 Experiments on KTH Database

The KTH dataset is one of the largest and most challenging datasets for human action recognition. This video database contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: S1: outdoors, S2: outdoors with scale variation, S3: outdoors with different clothes and S4: indoors with lighting variations [4]. Extracting silhouettes in this dataset is quite difficult because of the presence of shadows, severe jitter, lighting variations, camera movement and zoom. As mentioned in [71, 96] silhouettes are not very useful in the KTH dataset because of the difficulty in extracting them. This becomes clearer when we note that most of the works on this dataset including [4, 35, 56, 61, 97, 98] are based on local features extracted directly from the raw video, avoiding the difficulties in foreground extraction. Furthermore most of the literature dealing with silhouettes for action recognition does not report results on the KTH dataset. In [71], silhouettes have been extracted from a few of the videos of KTH from which a reliable foreground can be extracted, i.e., 36 sequences of walk and 32 sequences of run for a total of 68 sequences out of 2950 video clips. They have only performed a cross dataset test using the extracted video clips as the test samples and the Weizmann dataset for training.

Despite the stated unsuitability of silhouette-based methods for the KTH dataset, we have experimented with it for those actions for which a reasonably clean edge image representing the silhouette can be obtained. These actions are the in-place actions in the dataset, viz., boxing, hand clapping and hand waving. We manually draw a bounding box to contain the subject and assume that the position of the bounding box remains the same throughout the action in all the frames (except for S2 in which the position and size of bounding box changes\(^1\)). We apply the Canny edge detector inside the bounding box that results in a silhouette of the subject. It is these edges that are now embedded in the low-dimensional space. All windowed frames are normalized to 80 × 48 pixels. Examples of computed edge maps for in-place actions are illustrated in figure 3.19. We see that these edge maps, resulting from manual bounding boxes, are sometimes corrupted and noisy. So estimating the action period is not straightforward, thus we use MHD in this section. Table 3.6 compares the performance of our method with other approaches for the

\(^1\)In fact, S2 is only the scale variation of S1, so the normalized foregrounds of S2 will be similar to those for S1.
three in-place actions. Clearly our method outperforms every method reported except the state of the art of Lin et al. [67]. Thus, we believe that by using more sophisticated edge detectors such as Berkeley Pb detector [102] or Line Segment Detector of von Gioi [103], the proposed method could achieve higher recognition rates. This is proposed to be part of the future work. In this work, our objective has been to introduce a new faster embedding method for silhouettes.

### 3.6 Conclusion

We have proposed a novel embedding method for action recognition that gives the most discriminant embedding based on SCD as the distance measure between sequences of silhouettes. Actions are modeled by sequences of key poses chosen equidistantly during one action period. The poses are embedded into the learned subspace and compared in the projected space by either SCD or MHD. Several experiments are carried out on three popular datasets to demonstrate the efficiency and power of the proposed embedding. In addition to obtaining results comparable to state of the art on all datasets, our method
is outperforming other common dimension reduction methods in both the accuracy and time. Moreover, the method is verified to be robust to additive noise and tolerant to occlusions and various deformations. Also it is view-invariant up to promising extents.
Chapter 4

Learning Semantic Visual Codebook for Action Recognition by Embedding into Concept Space

4.1 Introduction

Action recognition using silhouettes [3] or optical flow [24] usually encounters difficulties when dealing with nonuniform background, severe camera jitter and noise. In contrast, local spatio-temporal features [4, 35] are fast and easy to extract and are also reliable. In this chapter, these features have been integrated into the Bag of Words (BOW) framework in which a visual vocabulary of the features is constructed by vector quantization of the features. Vector quantization involves clustering of the features by k-means and selecting the cluster centers as visual words. Recognition is done using histograms of visual words by means of a proper classifier. One of the important drawbacks of the standard BOW approach is that clustering of the features is only based on their appearances and not on their semantic relations. Using semantic relations improves recognition accuracy.

There have been some works which incorporate semantic relations into the standard BOW approach. For instance generative methods such as Probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation have been used [104] to build a model for each category and fit the query to one of the models in an unsupervised framework. The unsupervised nature of these methods limit their efficiency. Moreover the number of topics is equal to the number of categories, which is also a limiting factor. On the other hand discriminative methods try to construct a semantic visual vocabulary and use it
with a proper classifier [105, 106]. Among the discriminative methods, is the remarkable work of Liu and Shah which maximizes the mutual information between visual words and images [107] or videos [105]. Starting from singleton clusters, the algorithm iteratively merges the pair of clusters which result in the minimum loss in mutual information. The procedure continues until a threshold in either mutual information or number of clusters is achieved. This method is useful in finding the optimum number of clusters but the formed clusters do not necessarily represent the synonym words or semantic topics. In another work, Liu et al. [106, 108] use Diffusion distance to construct a semantic visual vocabulary. In contrast to geodesic distance, which is based on the shortest path, diffusion distance is measured along all paths between two points and thus it is not sensitive to noise. However considering the connectivity between points results in inappropriate distance in the presence of polysemes. For example, suppose word B has 2 different meanings. Assuming that in the first meaning words A and B are synonyms and in the second meaning B and C are synonyms, then based on diffusion distance, A and C are synonyms, which is incorrect.

Considering these drawbacks, in this chapter we propose a framework for constructing a semantic visual vocabulary via computing a rich semantic space which we call the concept space. The concept space is obtained by latent semantic models or canonical correlation analysis (CCA). The visual words are embedded into the concept space to form meaningful clusters which represent the semantic topics. Consequently the formed histograms based on these semantic clusters are discriminative between classes.

As opposed to generative methods which do not use the category labels, our proposed framework incorporates a classifier which is trained on the training histograms. Moreover in our method the number of topics can be more than the number of categories which results in finer details in modeling the semantics. Also, by using pLSA in constructing the concept space, the problem of polysemy is handled. We have tested our proposed method on the challenging KTH human action database [4] and achieved promising results.

4.1.1 Overview of the proposed framework

Figure 4.1 shows the flowchart for constructing the semantic visual vocabulary. First

---

1Polysemes are the words which have more than one meaning. For instance the word *table* can be referred to as *a piece of furniture* or *an arrangement of data.*
the raw features are extracted from the spatio-temporal volume of the video data. Then the initial vocabulary is built by clustering the raw features using k-means clustering and choosing the set of centers of clusters as the codebook or the visual words. We quantize the raw features based on the initial vocabulary to construct the word-video co-occurrence matrix. In this representation, each word is represented as its frequency in the training videos. Subsequently the visual words are embedded into the obtained concept space. The projected words in the concept space which we believe to be more semantically discriminant are further clustered by k-means to obtain the semantic visual vocabulary. Figure 4.2 indicates the major steps for training phase in our algorithm. The constructed semantic vocabulary is used to quantize the raw features and represent each video as a histogram of words. These representations are used to train the SVM classifier.

4.2 Related Work

Since clustering in BOW framework is done based on appearance of features rather than their semantic relations, there have been several attempts to introduce semantic relations for more efficient discrimination. These fall into two categories: Generative and Discriminative methods. Generative methods usually involve hidden variables and a video is represented by a mixture of hidden concepts. Recognition is done in an unsupervised framework using either pLSA [104,109] or Latent Dirichlet Allocation [104,110].
maximum achievable recognition accuracy in these methods is limited due to their unsupervised nature. On the other hand, discriminative methods are only based on observed variables and generally use a classifier. Vogel and Schiele [111] have defined concept classes like sky, sand and sea to label image regions. They use color and texture features in order to represent image regions and classify them to concept classes. Subsequently an occurrence vector of concept classes are formed for each image which is used for recognition. The obvious drawback is the large amount of manual work needed for annotation. Randomized clustering forests is an effective method which has been used for image classification [112]. In this method, a set of decision trees is built based on class labels. Then, visual words are assigned to each leaf of the tree. Based on a threshold for number of leaves, a bottom-up pruning process is performed. Randomized forest is fast compared to k-means, but it tends to overfit especially in noisy situations. In Liu et al.’s work [106, 108], briefly mentioned in the Introduction, diffusion maps are used to build a semantic visual codebook. In this method, visual words are represented by nodes of a graph and described by their pointwise mutual information. The similarity between visual words is reflected in the weights between the corresponding nodes. Diffusion map is used to embed the visual words into a subspace in which the Euclidean distance is equal to diffusion distance. Simple low-level visual features have been exploited by Wang and Li [113]. They capture motion by simple frame differencing. A set of key poses is then selected by clustering the data. Action is represented as a sequence of key poses augmented with discriminative features. Weighted-sequence distance is used between sequences wrapped as a kernel classified by SVM. Compared with their work which extract features frame-wise, our proposed method extracts features from the whole video volume. Moreover our work compares actions based on the distances between semantically learned histograms while their method is based on distances between sequences of key poses.

4.3 Concept Space

In the standard BOW framework, an initial vocabulary is obtained by applying k-means on the extracted visual features. The obtained vocabulary represents all features but
Chapter 4. Learning Semantic Visual Codebook for Action Recognition by Embedding into Concept Space

the formed clusters do not represent semantic topics. Thus the histograms formed from them are not semantically discriminative. We require a space which infers the semantic relations. In order to incorporate the semantic relations we use latent semantic models (Latent Semantic Analysis and probabilistic Latent Semantic Analysis) as well as Canonical Correlation Analysis. We detail these approaches in the following sections. Both these methods use the word-video co-occurrence matrix in order to extract the semantic relations between words.

4.3.1 Embedding into concept space using Latent Semantic Analysis

Latent Semantic Analysis (LSA) [114] originally used in text mining applications, is the factorization of word-video co-occurrence matrix into linear subspaces of words and videos denoted as latent semantic spaces. Given a collection of $N$ videos $D = \{d_1, ..., d_N\}$ consisting of $M$ visual words from a codebook $W = \{w_1, ..., w_M\}$, let $X$ be the word-video co-occurrence matrix defined as $X = (n(w_i, d_j))_{ij}$, where $n(w_i, d_j)$ infers the number of occurrences of word $i$ in video $j$. The rows and columns of $X$ is called word and video vectors respectively. The word vectors reveal the semantic relations of words, since semantically synonymous words occur in similar documents. However, the word vectors are sparse and so their correlation may not be so representative of their semantic relations. Their likelihood might be small since they are not using the exact same words. Therefore, we need to find the reduced dimensional space. Decomposing $X$ using SVD as $X = U \Sigma V^T$ gives orthogonal matrices $U$ and $V$ as well as the diagonal matrix of singular values $\Sigma$. The columns of $U (V)$ span the space of words/videos. Picking the $L$ largest singular values and the corresponding singular vectors will lead to the rank $L$ optimal representation in the sense of Frobenius norm:

$$X \approx U_L \Sigma_L V_L^T.$$  \hspace{1cm} (4.1)

The correlation of words based on word vectors is computed as $XX^T$. Using the rank $L$ approximation we get:

$$XX^T \approx U_L \Sigma_L V_L^T V_L \Sigma_L^T U_L^T = U_L \Sigma_L \Sigma_L^T U_L^T.$$  \hspace{1cm} (4.2)
Rows of $U_L\Sigma_L$ are a good representation of rows of $X$ (words) in the sense that they approximate the correlation between words and consequently their similarities. Thus the $i^{th}$ row of $U_L\Sigma_L$ denoted as $\hat{t}_i$ which is no longer sparse is a representation of word $i$ in the concept space with dimension $L$.

### 4.3.2 Embedding into concept space using Probabilistic Latent Semantic Analysis

Probabilistic Latent Semantic Analysis (pLSA) \[115\] has been motivated from a statistical model called Aspect Model \[116\] or Aggregate Markov Model \[117\] in the context of language modeling. pLSA assumes a latent variable $z_l$ for each occurrence of a word $w_i$ in a document $d_j$. $w_i$ and $d_j$ are observed while $z_l$ is latent. In \[104,115\] $z_l$ represents the categories. However we use them as semantic topics, so their number is typically much more than the categories. The probability of occurrence of a word in a particular video $d_j$ is denoted as $P(d_j)$. $P(w_i|z_l)$ denotes the probability of a particular word $w_i$ conditioned on the topic $z_l$. Similarly the video-specific probability distribution of topics is denoted as $P(z_l|d_j)$. The generative model describing the co-occurrences of words and videos can be defined as the following: The video is selected with probability $P(d_j)$, then a topic is chosen with probability $P(z_l|d_j)$. Subsequently a word is picked based on $P(w_i|z_l)$. Discarding the unobserved latent variable, the probability of observation $(w_i,d_j)$ can be computed as:

$$P(w_i,d_j) = P(d_j)P(w_i|d_j).$$  \hspace{1cm} (4.3)

It is assumed that $w_i$ and $d_j$ are independent conditioned on the latent topics. Therefore we can marginalize over latent topics to compute the conditional probability $P(w_i|d_j)$:

$$P(w_i|d_j) = \sum_{l=1}^{L} P(w_i|z_l)P(z_l|d_j),$$ \hspace{1cm} (4.4)

Maximum likelihood can be used to find the parameters of the model. The log-likelihood function to be maximized with respect to all probability masses is:

$$\mathcal{L} = \sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i,d_j) \log P(w_i,d_j)$$

$$= \sum_{j=1}^{M} n(d_j) \log P(d_j) + \sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i,d_j) \log P(w_i|d_j),$$ \hspace{1cm} (4.5)
Chapter 4. Learning Semantic Visual Codebook for Action Recognition by Embedding into Concept Space

where \( n(d_j) = \sum_{i=1}^{M} n(w_i, d_j) \). The maximum likelihood is obtained by the Expectation-maximization (EM) in which two steps are alternatively performed: 1-an expectation (E) step for computing posterior probabilities for the topics; 2-a Maximization (M) step for updating the parameters based on the expected complete data loglikelihood. For the E-step, the following can be obtained from equation (4.4) based on Bayes’ rule:

\[
P(z_l|w_i, d_j) = \frac{P(w_i|z_l)P(z_l|d_j)}{\sum_{i=1}^{M} P(w_i|z_l)P(z_l|d_j)}.
\]  

(4.6)

For the M-step we have to maximize the complete log-likelihood \( E[L_c] \). \( P(d_j) \) is proportional to \( n(d_j) \), so we maximize the nontrivial part which is:

\[
E[L_c] = \sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i, d_j) \sum_{l=1}^{L} P(z_l|w_i, d_j) \log[P(w_i|z_l)P(z_l|d_j)].
\]

(4.7)

In order to enforce the normalization constraints for probability masses, Lagrange multipliers are added:

\[
H = E[L_c] + \sum_{i=1}^{L} \tau_i (1 - \sum_{i=1}^{M} P(w_i|z_l)) + \sum_{j=1}^{N} \rho_j (1 - \sum_{l=1}^{L} P(z_l|d_j)).
\]

(4.8)

The M-step update equations are obtained by eliminating the Lagrange multipliers as:

\[
P(w_i|z_l) = \frac{\sum_{j=1}^{M} n(w_i, d_j) P(z_l|w_i, d_j)}{\sum_{m=1}^{M} \sum_{j=1}^{N} n(w_m, d_j) P(z_l|w_m, d_j)},
\]

(4.9)

\[
P(z_l|d_j) = \frac{\sum_{i=1}^{M} n(w_i, d_j) P(z_l|w_i, d_j)}{n(d_j)}.
\]

(4.10)

The two steps (E-step and M-step) are alternatively performed until a stopping criteria is met. In the original pLSA algorithm [104, 115], after learning the \( P(w|z) \), in order to find \( P(z|d) \) for the new video, the query video is projected on the simplex which is spanned by \( P(w|z) \). Thus in order to find the mixing coefficients \( P(z_l|d_{est}) \), the KL divergence between empirical \( \hat{P}(w|d_{est}) \) and \( P(w|d_{est}) = \sum_{l=1}^{L} P(w|z_l)P(z_l|d_{est}) \) are minimized.

But we do not need \( P(z_l|d_{est}) \). We use pLSA only to find the topic-specific distribution of word \( P(w_i|z_l) \) which is equivalent to the \( l \)th dimension of \( \hat{i} \) in LSA framework. So we embed the word \( w_i \) into concept space by:

\[
\hat{i} = \left[ p(w_i|z_1) \ p(w_i|z_2) \ ... \ p(w_i|z_L) \right]^T.
\]

(4.11)
Note that $L$ is the dimension of the concept space or the number of possible topics which is typically much more than the number of categories. In fact by allowing the number of topics to go beyond the number of classes, we allow finer and more detailed concepts.

In embedding into concept space using LSA, the word can be embedded into a single point equivalent to a single meaning, so there is no freedom for having more than one meaning. On the other hand, in pLSA, the word $w_i$ having two different meanings in the two videos $d_i$ and $d_j$ can be associated with different topics or $\arg\max p(z|d_i, w)$ can be different from $\arg\max p(z|d_j, w)$ [115], [118]. So pLSA can handle polysemy. But LSA can be performed faster.

### 4.3.3 Embedding into concept space using Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) was first proposed by Hotelling [82] and has been used in some computer vision applications. Hardoon et al. [119] have used kernel CCA to find a common semantic space between images and their corresponding text. This common semantic space allows comparison of image contents with text and retrieval of images based on a text query. In another application, Kim et al. [83] have used CCA for comparing sets of images for object recognition. The image sets include variations of viewpoint and also lighting condition. They have used CCA as the angle between linear spaces or principal angles. Inspired by Linear Discriminant Analysis they have proposed a linear discriminant function which minimizes the within-set canonical correlations while maximizing the between-set canonical correlations.

Given a pair of vector sets, CCA finds the direction for each set such that the projection of the vectors onto these directions have maximal correlation. Assume a bivariate random vector of the form $(x, y)$. Suppose we are provided with the sample instances $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. Let $S_x = \{x_1, \ldots, x_n\}$ and $S_y = \{y_1, \ldots, y_n\}$. Consider a direction $w_x$ onto which we project $x$, where the projection is given by $\langle w_x, x \rangle$. Similarly, we project $y$ onto $w_y$. The $x$ coordinate of the projected sample will be $S_{x,w_x} = \{\langle x_1, w_x \rangle, \ldots, \langle x_n, w_x \rangle\}$ and similarly the $y$ coordinate is $S_{y,w_y} = \{\langle y_1, w_y \rangle, \ldots, \langle y_n, w_y \rangle\}$. In Canonical Correlation Analysis our aim is
to find directions $w_x$ and $w_y$ such that the correlations of the projected vectors are maximized [119]. In other words we have to maximize the following objective function:

$$\rho = \max_{w_x, w_y} \text{Corr}(S_{x,w_x}, S_{y,w_y})$$

$$= \max_{w_x, w_y} \frac{\langle S_{x,w_x}, S_{y,w_y} \rangle}{\|S_{x,w_x}\| \|S_{y,w_y}\|}. \quad (4.12)$$

If the empirical expectation of the function $f(x, y)$ is $\hat{E}[f(x, y)] = \frac{1}{m} \sum_{i=1}^{n} f(x_i, y_i)$, then we can rewrite equation (4.12) as:

$$\rho = \max_{w_x, w_y} \frac{\hat{E}[\langle w_x, x \rangle \langle w_y, y \rangle]}{\sqrt{\hat{E}[\langle w_x, x \rangle^2] \hat{E}[\langle w_y, y \rangle^2]}}$$

$$= \max_{w_x, w_y} \frac{\hat{E}[w_x^T x y^T w_y]}{\sqrt{\hat{E}[w_x^T x x^T w_x] \hat{E}[w_y^T y y^T w_y]}}$$

$$= \max_{w_x, w_y} \frac{w_x^T \hat{E}[x y^T] w_y}{\sqrt{w_x^T \hat{E}[x x^T] w_x w_y^T \hat{E}[y y^T] w_y}}. \quad (4.13)$$

Now if we write equation (4.13) in terms of covariances we obtain:

$$\rho = \max_{w_x, w_y} \frac{w_x^T C_{xy} w_y}{\sqrt{w_x^T C_{xx} w_x w_y^T C_{yy} w_y}}, \quad (4.14)$$

where $C_{xx}$ and $C_{yy}$ are the within-set covariance matrices and $C_{xy}$ is the between-set covariance matrix. Note that the solution to equation (4.14) is not affected by rescaling $w_x$ and $w_y$ either together or independently. As the choice of rescaling is arbitrary, the solution to equation (4.14) can be found by obtaining the solution to the following constrained optimization problem:

$$\rho = \max_{w_x, w_y} w_x^T C_{xy} w_y \quad \text{s.t.} \quad \begin{cases} w_x^T C_{xx} w_x = 1 \\ w_y^T C_{yy} w_y = 1 \end{cases} \quad (4.15)$$

Optimizing equation (4.15) while assuming $C_{yy}$ is invertible, $w_y$ is obtained by [119]:

$$w_y = \frac{C_{yy}^{-1} C_{yx} w_x}{\lambda} \quad (4.16)$$

Finally $w_x$ is calculated based on the following generalized eigenvalue problem:

$$C_{xy} C_{yy}^{-1} C_{yx} w_x = \lambda^2 C_{xx} w_x \quad (4.17)$$
Livescu et al. [120] have shown that given two different representations of the same signal which are polluted with (high dimensional) additive noise, CCA yield a subspace in which the noise covariance is reduced relative to the signal covariance. In particular, assume a mixture of distributions with each distribution pertaining to a hidden semantic topic. We have samples from two different representations. The sample variables are denoted by \( x \) and \( y \) related to the first and second representation respectively. The samples from the variable \( x \) are derived from the mixture of \( L \) Gaussians \( (Z^x_1, ..., Z^x_L) \). To generate a sample, first a distribution is chosen. Subsequently a sample is picked according to the chosen distribution. Chaudhuri et al. [121] have proved that provided two conditions are satisfied, we can recover the subspace containing the distribution means of both representations using their CCA. Similar statements hold for variable \( y \) as well. The two conditions for subspace recovery are:

(i) given the distribution space (topic), the two representations are uncorrelated.

(ii) the rank of the between-set covariance matrix \( C_{xy} \) is at least \( L - 1 \).

If the means in one of the representations are separated well enough, we can cluster the samples from the other representation in the resulting CCA space correctly with high rates [121]. The probability of correct clustering is dependent on the number of samples and the smallest distance between the means. Larger distances between means increases the probability of correct clustering.

In our case, we choose two representations to model the visual words: the raw feature representation and the semantic representation which is the word-video co-occurrence vector. The semantic representation of words is sparse and high-dimensional (dimension is equal to the number of training videos which is typically 2950 in our experiments). But the actual number of topics corresponding to synonym words is much less. Also two synonym words may not have the same representations, which indicates that our representation has some sort of noise. On the other hand, our feature representation of words explained later in section 4.4 provides a good discrimination, which is why the standard BOW framework using these features gives satisfying results.

Next, we check whether the two conditions for subspace recovery hold for our case. The first condition is certainly satisfied since the two representations are uncorrelated,
Chapter 4. Learning Semantic Visual Codebook for Action Recognition by Embedding into Concept Space

Figure 4.3: Constructing the semantic visual vocabulary.

given the topics. The rank condition requires that the rank of CCA matrix to be at least $L - 1$. In our particular problem, since the number of clusters or semantic topics cannot be determined, this condition can be satisfied by assuming the proper number of topics (the number of topics should always be at most rank of CCA matrix plus one). Based on the theory, we can recover the subspace containing the means given that the visual words are sufficiently separated in the raw-feature space. Although in our case the semantic topics are hidden and unknown, by assuming them to follow a statistical distribution similar to speakers or Wikipedia documents in [121], we use the conditions that are valid here. We verify that the proposed method is efficient through experiments in section 4.5.

Figure 4.3 shows the flowchart for constructing the semantic visual vocabulary using CCA. The initial vocabulary is built as before and used to construct the word-video co-occurrence matrix. The two views of visual words, i.e. the semantic representation, $y$ or co-occurrence of word in training videos and the raw-feature description, $x$ are used to find the subspace in which the two views are maximally correlated. The projected words given by $S_{y,w_y}$ which we believe to be more semantically discriminative are further clustered by k-means to obtain the semantic visual vocabulary.

4.4 Feature Extraction

We use the dense feature extractor proposed by Dollar et al. [35] which has shown to perform better compared to sparse interest point detectors like [4]. A response function is defined over the sequence which incorporates Gaussian filter in the spatial domain and Gabor filter in the temporal domain. Interest points are detected at the places in which the response function has local maxima. Fixed-size cuboids are extracted around
detected interest points. Some examples of interest points detected are illustrated in figure 4.4. The patches shown correspond to the extracted cuboids in the sequences. Each cuboid is described by flattened gradients. The dimension of these descriptors are reduced by PCA to 100 to form the feature vectors.

4.5 Experiments

In this chapter we perform experiments on the KTH action dataset [4] since it is one of the largest and most challenging datasets for human action recognition. It contains variance in appearance as well as scale and illumination. It consists of 6 actions - boxing, hand clapping, hand waving, jogging, running and walking - performed by 25 subjects under 4 different scenarios - outdoors, outdoors with scale changing, outdoors with different cloth and indoors with lighting variations. There are a total of almost 1200 video clips in this database. We use the leave-one-out cross validation technique to test the performance, i.e., each time we train with videos of 24 persons and use the videos pertaining to the remaining person for test and report the average of the recognition results. Support Vector Machine (SVM) with Histogram Intersection kernel is used as the classifier. More information about Histogram Intersection Kernel is given in section 5.4.1. We choose the size of the initial codebook as 1500 for all the experiments.

4.5.1 Experiments using Latent Semantic Space

One of the advantages of the proposed method is that it allows the number of topics to be varied, in contrast to pLSA using unsupervised framework where the number of topics is constrained to be the same as the number of classes. Figure 4.5 (a) and (b) show the influence of number of topics $L$ on the recognition accuracy with LSA and pLSA as the embedding method. The experiments have been performed using three different semantic vocabulary sizes, $K_f$. As the number of topics is increased from $L = 6$, which is the number of classes, the recognition rate increases since the increased number of concepts enables better discrimination between topics. However, after around $L = 30$ topics, the recognition accuracy decreases. This is mainly because adding more dimensions to the concept space implies further division into semantic units that are not meaningful. This
Figure 4.4: Some examples of detected interest points for six actions of KTH dataset overlaid on sample images. The patches indicate extracted cuboids in sequences. As seen interest points are mainly extracted from parts which are involved in the main motion.
phenomenon occurs at $L = 50$ for pLSA with $K_f = 400$. The recognition accuracy has a variance of about 2%-3% as $L$ varies.

In order to determine the efficiency and discriminative power of the proposed method compared to the classic BOW framework which uses the initial vocabulary to build the query histogram without embedding into any concept space, we compared the accuracy of our method with the classic framework with the same size of the final codebook. The results are illustrated in figure 4.6(a) for $L = 30$. The proposed method outperforms the classic BOW framework for all vocabulary sizes shown, illustrating that our approach is discriminative and effective in recognizing actions. Also the variance of recognition accuracy for our method in different vocabulary sizes is about 2%. Hence, our method is not sensitive to the size of the codebook. For small vocabulary sizes, pLSA outperforms
Chapter 4. Learning Semantic Visual Codebook for Action Recognition by Embedding into Concept Space

LSA by a small margin due to its ability to handle polysemy. But as the vocabulary size increases, LSA performs better than pLSA probably because the larger codebook size brings in more details and compensates for the polysemy effect. However, pLSA takes into account every possible meaning of a word, even the rare ones, which results in confusion in larger vocabularies, thus reducing the accuracy.

The best recognition accuracy of our method is 93.94% which is achieved using pLSA with $L = 50$ and $Kf = 400$. Figure 4.6(b) shows the related confusion matrix. Most confusions are between jogging and running due to the strong similarities between these actions which is hardly distinguishable even for humans. Moreover, our method has successfully recognized walking despite its similarity to jogging and running.

LSA always has a smaller training time compared to pLSA due to the time consuming iterative EM process for pLSA compared to the straightforward SVD in LSA. Assuming we have the initial vocabulary, the mean training time for LSA is 62 seconds while for pLSA is 4261 second. Once the concept space has been learned the testing time is almost the same for both methods. The testing time which includes embedding into concept space and SVM prediction is 0.54 seconds for LSA and 0.71 seconds for pLSA.

### 4.5.2 Experiments using Canonical Correlation Space

The CCA is obtained using two representations of the visual words: the raw feature form proposed by Dollar et al. [35], $x$ and the representation including the co-occurrence of words in the training documents, $y$. We use $S_{wy}$ as the projected words to further cluster them by k-means.

The dimension of the CCA space is the minimum dimension of two representations. In our particular case, the raw features have a dimension of 100 and there are totally 2295 training sequences, so the CCA space at most have a dimension of 100. We can still reduce the dimension to include only the highest canonical correlation directions. This way it is also easier to further cluster the features, since we are dealing with lower dimensional vectors. Figure 4.7 shows the effect of changing the dimension of CCA space on the recognition accuracy for tree sizes of the final vocabulary, $Kf$. Two factors should be noted in this figure. First the amount of information is increased by using more dimensions in CCA space which may result in increasing the accuracy. On the other
hand, vectors with fewer dimensions are easier to be clustered using k-means which helps in improving the accuracy. Still the first dimension alone has most of the information needed, which has resulted in higher recognition as seen in the figure.

In order to discover the efficiency of using CCA space over the classic BOW which uses initial vocabulary to build the histograms, we have compared our method with the standard BOW with the same sizes of final vocabulary, $K_f$. The results are shown in figure 4.8. The values used are the best results over the dimension range. As seen in the figure, our method outperforms the classic BOW for almost all the codebook sizes which shows that our approach is efficient and discriminative in recognizing actions. Moreover the variance of recognition accuracy over different sizes of vocabulary is 1.5 % which shows that our method is not so sensitive to size of vocabulary. As in figure, larger vocabulary sizes generally bring more details and increase the accuracy.

The best Recognition accuracy obtained by our method is 93.39 % which is achieved in $K_f = 700$. Figure 4.9 illustrates the confusion matrix regarding this result. According to figure, most confusions are among jogging and running which are very similar. Moreover,
we have successfully distinguished walking despite its strong similarities with jogging and
running.

Assuming the initial vocabulary is provided, the mean training time for CCA is 4
seconds which is in the range of LSA since they both rely on eigenvalue decomposition.
The mean testing time is 0.52 seconds which is almost similar to LSA and pLSA.

**4.5.3 Comparison with Other Methods**

In order to illustrate the efficiency of the proposed method, we have compared our results
with other results in action recognition on the KTH database in table 4.1. The accuracy
shown is based on all scenarios combined not an average of all scenarios. The proposed
method outperforms methods using semantic visual vocabulary, like Liu(MMI) [105] and
Liu(DM) [106] and also some other important results reported on the KTH dataset. The
method of Lin et al. [67] is slightly better than ours when LSA is used. Their method is
based on information from both shape and motion, while our approach is solely based on
motion. We have obtained the best result of 93.94% accuracy using BOW model without
any spatial or temporal information. Hence, we have not compared our results with
those that use structural information e.g. section 3 of [105], which reports an accuracy
of 94.16%.

![Confusion Matrix](image)
Table 4.1: Comparison with recently reported results for KTH dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours(pLSA)</td>
<td>93.9%</td>
</tr>
<tr>
<td>Ours(CCA)</td>
<td>93.4%</td>
</tr>
<tr>
<td>Lin [67]</td>
<td>93.4%</td>
</tr>
<tr>
<td>Ours(LSA)</td>
<td>93.3%</td>
</tr>
<tr>
<td>Schindler [58]</td>
<td>92.7%</td>
</tr>
<tr>
<td>Liu(DM) [106]</td>
<td>92.3%</td>
</tr>
<tr>
<td>Liu(MMI) [105]</td>
<td>91.3%</td>
</tr>
<tr>
<td>Gilbert [99]</td>
<td>89.9%</td>
</tr>
<tr>
<td>Rapantzikos [61]</td>
<td>88.3%</td>
</tr>
<tr>
<td>Wang [113]</td>
<td>87.8%</td>
</tr>
<tr>
<td>Nowozin [100]</td>
<td>87.0%</td>
</tr>
<tr>
<td>Niebles [104]</td>
<td>83.3%</td>
</tr>
<tr>
<td>Dollar [35]</td>
<td>81.2%</td>
</tr>
<tr>
<td>Li [101]</td>
<td>77.6%</td>
</tr>
<tr>
<td>Schuld [4]</td>
<td>71.7%</td>
</tr>
<tr>
<td>Ke [70]</td>
<td>63.0%</td>
</tr>
</tbody>
</table>

4.6 Conclusion

We have proposed a novel method for introducing semantic relations in the BOW framework for the purpose of human action recognition. Latent semantic models (LSA and pLSA) as well as CCA are used to project the visual words into a concept space in which the distances reveal semantic relations. The projected words are believed to be more semantically distributed. The embedded visual words are further clustered to capture the semantic units and construct the semantic visual vocabulary. The histograms which are computed based on the semantic visual vocabulary are more discriminative. Our results verified that the proposed semantic visual vocabulary is efficient and discriminative in recognizing human actions in BOW framework.
Chapter 5

Efficient Combination of View-dependent Histograms for Multi-view Human Action Recognition

In this chapter we address occlusion in recognizing human actions. In the previous two chapters, we classify human actions from a single view only. Recognizing actions from a single viewpoint is confronted by limitations like self occlusions or occlusions by other objects or humans. However, recognition using multiple viewpoints whenever available can help us overcome occlusion. How to efficiently incorporate information from multiple viewpoints in a single framework is the main concern in this chapter. We propose to extract spatio-temporal features from each viewpoint and construct histograms with different codebook sizes. The similarity between obtained histograms are computed using HIK kernel as well as RBF kernel with $\chi^2$ distance. Finally all the computed kernels are linearly combined with the proper weights which are determined through an optimization process. For higher efficiency we use a one versus one binary classification scheme. A separate set of optimum weights are obtained for each binary classifier. Several experiments are performed to verify the efficiency of the whole method as well as each constitutive part.
Chapter 5. Efficient Combination of View-dependent Histograms for Multi-view Human Action Recognition

5.1 Introduction

Although video describes a two-dimensional space which changes in time, actions truly occur in the three-dimensional world space. So like any other projections, we may lose important information in describing a truly three-dimensional phenomenon in two-dimensional space observed by a camera. One of the obvious situations when action recognition from a single camera fails is when the subject is being occluded by an object. Also unavoidable self-occlusions may hide some parts of the action. This motivates observation of an action with more than one camera in order to get a more complete picture of the action.

Some researchers have used multiple cameras to obtain a 3D representation of the subject using shape from silhouette and call it a visual hull [12, 122]. Subsequently, features are extracted from the visual hulls for recognition purpose. But in order to build a reliable visual hull, a sufficient number of views should be available. In fact in carving a visual hull from a number of views, some information is lost and visual hull becomes an approximation for the true 3D model. On the other hand, to build the visual hulls, silhouettes are needed in advance. Extracting foregrounds is not so straightforward especially in the cases of complicated backgrounds or moving cameras. Due to these disadvantages, we do not use visual hull for multi-view action recognition. Instead, we propose to extract features from each viewpoint separately and combine them efficiently such that useful information is reinforced and redundant features are attenuated. We extract local features from each view, which is easy to extract and does not require segmentation. Methods using visual hulls [12, 122] need the entire video sequence to be processed before deciding the category of action. In the proposed method, the video can be segmented into multiple small sequences and a decision can be made about each segment while video is running; thus it is more suitable for real-time applications. Peng and Qian [122] use a Hidden Markov process to model the order of transition between states or frames. We use a simple Bag of Word (BOW) model which is obviously orderless. Here we argue that only the distribution of features is sufficient to efficiently describe an action. So we can use the BOW framework which is much more easier to train or test and can be used with classifiers such as SVM.
Computer Vision researchers have combined different features for obtaining high robustness and accuracy for object (or scene) recognition [123–127]. All these methods use a number of heterogeneous features which aim at different aspects of the object (or scene). For instance, Gehler and Nowozin [124] use shape, color and texture features as well as features like HOG, clustered HSV values, SIFT features on the foreground region and SIFT features on the foreground boundary for the task of flower recognition. Extracting all these features may not be so straightforward and will consume much time and memory. In fact, feature extraction can be the most time consuming step in the whole recognition process. Methods which use fewer number of features but concentrate more on modeling may be more beneficial for real-time scenarios. In this work, we do not incorporate many heterogeneous features, but simple local features which are easily extracted without the prior need for segmentation. Instead of extracting many different features, we focus on computing different models using different codebooks and functions from the same set of features and combining them efficiently.

In order to achieve multi-class recognition from binary classification, we choose one-versus-one classification scheme over one-versus-all, since one-versus-one has more precision. Moreover when a category is being added later on, we do not need to re-train the whole system all over again. Each binary classification is done using a binary SVM classifier. We model the same video with different histograms obtained from two local features and two vocabularies. The distance between histograms are measured using HIK (Histogram Intersection Kernel) as well as RBF (Radial Basis Function) kernel with $\chi^2$ distance. We use an efficient interleaved optimization strategy to learn the optimum weights for the multiple kernels. The obtained optimum weights score each kernel based on its ability to discriminate between two different categories. This is very useful in the multi-class recognition scenarios. In other words, although some viewpoints are generally more informative and provide more details in total, this is not always true for all pairs of action categories. For instance it is usually assumed that the side view is generally better than the top view. But in deciding between actions cross arms and check watch, the actions are almost the same from the side view, but distinctive from the top view. Figure 5.1 shows examples of images from these actions in the two viewpoints. As seen the two actions are almost the same from the side, however from the top viewpoint one
Figure 5.1: example images from actions *cross arms* and *check watch* from side view as well as top view.
hand is visible in check watch while both hands are visible in cross arms. Our method helps to learn the discriminative views in differentiating between each pair of actions. By generating various models (histograms) for representing the action, comparing them in different spaces using different distances and efficiently combining them together we have achieved the state of the art accuracy on the challenging multi-view IXMAS dataset.

5.1.1 Overview of Approach

Figure 5.2 shows the flowchart for the proposed method. As seen, from each view, two different kinds of local features are extracted. One of the features is obtained by applying separable linear filters on space and time domains in order to get keypoints throughout the video. The cuboid containing the keypoints are described by applying brightness gradients followed by PCA [35]. Another local feature is also extracted by applying an space-time corner detector at multiple spatial and temporal scales and described by an HOG feature concatenated by HOF (Histogram of Optical Flow) [14]. The raw features are clustered by k-means into V and 2V number of clusters to generate two codebooks per feature. We use the codebooks to quantize the features extracted from each video in order to represent it with histograms. Subsequently HIK distance as well as $\chi^2$ distance between histograms are computed and inserted in linear and exponential functions respectively to form the fundamental kernels. These kernels are linearly combined with proper weights to form the final kernel used with the SVM binary classifier. The optimum weights for the fundamental kernels are obtained by a 2-norm non-sparse multiple kernel learning algorithm [128].

5.1.2 Organization

This chapter is organized as follows. Section 5.2 reviews some related works on multi-view human action recognition. Representing the video with histograms of words is described in section 5.3. The method for combining the histograms is detailed in section 5.4. Section 5.5 presents and discusses the experimental results. Finally we conclude the chapter in section 5.6.
Figure 5.2: The method for generating basic kernels $K_{c,f,v,k}$ from all the possible combinations of constitutive factors, namely, camera viewpoint ($c \in \{\text{View 1}, \ldots, \text{View C}\}$), feature type ($f \in \{\text{Separable Linear Filters, Space-time Corner Detector}\}$), codebook size ($v \in \{V, 2V\}$) and kernel type ($k \in \{\text{HIK distance in a linear function, Chi-Square distance in an exponential function}\}$). The basic kernel $K_{c,f,v,k}$ is weighted with $w_{c,f,v,k}$. $K$, the final kernel used, is the linear combination of basic kernels.

$k = \text{zeros}();$

\[ K = K + w_{c,f,v,k} K_{c,f,v,k}; \]
5.2 Related works

In this section we discuss methods which combine information from different views for action or gesture recognition. These methods generally fall into two categories: methods that combine multiple views into a 3D model to remove the view-dependent information and subsequently use the model to compare different actions [12, 122, 129, 130] and methods that use fusion techniques to combine information from different views [131].

Among the approaches using 3D model, Weinland et al. [12] introduce Motion History Volumes based on the idea of Motion History Images [2]. They use extracted silhouettes from different views to generate a 3D visual hull using the carving method. In this method the initial 3D volume is discretized into voxel points. Using the calibration matrices, each voxel point is projected onto the image planes of the cameras. Only the voxel points which are projected onto the foreground of all the cameras are considered to be part of the visual hull. The integration of all the visual hulls over time in a sequence is represented in the Motion History Volume. Subsequently Fourier transforms in cylindrical coordinates are extracted from the volumes as the motion descriptor. The descriptors are projected by PCA and classified based on the Mahalanobis distance from each class mean.

Visual hulls are also used in the work of Peng and Qian [122]. Voxel data are projected onto a low dimensional pose coefficient space using multilinear analysis. Eventually sequences of pose descriptors are represented and classified by hidden Markov models. Pehlivan and Duygulu [130] have divided visual hulls into horizontal layers. Each layer is coded by the enclosing circle. The properties of enclosing circles are used to describe each pose. Consequently the pose descriptors of all the frames in the sequence are combined into a motion descriptor which is used in a nearest neighbor framework.

The above approaches based on visual hulls need foreground extraction to be done in advance. This is not straightforward especially in the cases of complex backgrounds. Moreover in reconstruction of the visual hull from multiple views some information are lost. The more view incorporated, the closer the visual hull will be to the true 3D model but with the cost of higher computational complexity. Therefore some other researchers have proposed to fuse the views either in the feature-level or decision-level [131].
et al. [131] presents three fusion approaches for multi-view action recognition: (a) Best-view fusion: it trains a classifier for each camera independently. For each episode of the clip, the fusion process chooses the best view camera. Different criteria can be used to determine the best-view camera. The authors use the number of detected spatio-temporal features as their criterion. (b) Combined-view fusion: it simply concatenates all features from different cameras into a single feature to be classified. (c) Mixed-view fusion which uses the same classifier regardless of viewpoint. Best-view fusion has the highest accuracy and mixed-view the lowest. Our proposed method is considered as a smart way of feature fusion which automatically chooses the best-view for each pair of actions separately while combining all the other features with lower weights in an optimized manner, thus outperforming naive concatenation of features.

5.3 Histogram Representation of Video

In this section we describe how we generate different histograms from the video. These different histograms enable us to model the video with different criteria and in different details which lead to analyzing the video in greater depth. The different histograms are generated based on two local features - Separable Linear Filters [35] and Space-Time Corner Detector [14]- and also two sizes of codebook. The codebook is generated by clustering the training pool of features by k-means using Euclidean distance as metric. First we describe the local features used and their functionality, and then explain about choosing the two sizes for vocabulary. The two spatio-temporal local features have been chosen since they are simple to compute and do not need foreground segmentation in advance. They efficiently capture motion throughout the video volume. Moreover they have been successfully used in bag-of-words framework for the application of human action recognition [14,105,106].

5.3.1 Separable Linear Filters

Assuming a stationary camera, separable linear filters apply a Gaussian filter to the spatial domain and a quadratic pair of Gabor filters to the temporal dimension to obtain a response function as follows [35]:

\[ R = (I * g * h_{ev})^2 + (I * g * h_{od})^2, \]  

(5.1)
where \( g(x, y; \sigma) \) is the 2D Gaussian smoothing filter, and \( h_{ev} \) and \( h_{od} \) are the quadratic Gabor filters defined as 
\[
    h_{ev}(t; \tau, \omega) = -\cos(2\pi t \omega) \times e^{-t^2/\tau^2} \quad \text{and} \quad h_{ev}(t; \tau, \omega) = -\sin(2\pi t \omega) \times e^{-t^2/\tau^2}.
\]
The parameters \( \sigma \) and \( \tau \) represent the spatial and temporal scale of the detector respectively. \( \omega \) is always chosen as \( 4/\tau \). In order to handle multiple scales, one can run the detector at multiple spatial and temporal scales. For computational reasons we use only one scale and instead rely on the vocabulary to obtain robustness with respect to scale changes. In all the experiments we use \( \sigma = 2, \tau = 3 \). The space-time interest points are detected at local maxima of the response function. A cuboid is extracted around each keypoint with size of around six times the scale parameter along that dimension.

The extracted cuboids are described using its brightness gradients along different directions \( x, y \) and \( t \). Before computing the gradients, the cuboids are smoothed at different scales. The computed gradients at different directions and different smoothing scales are concatenated to form a vector. This vector is projected onto a lower dimensional space by Principal Component Analysis (PCA) to obtain the final descriptor. The size of the final descriptor in our experiments is 100.

### 5.3.2 Space-Time Corner Detector

The Space-Time Corner Detector is the extension of the Harris corner detector and is proposed by Laptev and Lindeberg [132]. A spatio-temporal second-moment matrix is computed by

\[
    \mu(0; \sigma', \tau') = g(0; s\sigma', s\tau') \ast (\nabla L(0; \sigma', \tau')(\nabla L(0; \sigma', \tau'))^T),
\]

where \( g \) is the Gaussian smoothing kernel with independent spatial and temporal scale values \( \sigma', \tau' \) and \( \nabla L \) is the space-time gradient and \( \ast \) denotes convolution. The final location of keypoints are determined by local maxima of the function 
\[
    Z = \det(\mu) - \rho \text{trace}^3(\mu), Z > 0.
\]
Instead of spatio-temporal scale selection as proposed by [132], we follow [14] and use multiple levels of spatio-temporal scales. This has shown to be effective for action recognition [14]. We choose spatio-temporal scales as \( \sigma'^2 = 4, 8, 16, 32, 64, 128, \tau'^2 = 2, 4 \). We use \( \rho = 5 \times 10^{-7} \).

Histograms of spatial gradients and optical flow are computed for each cuboid around the keypoints in order to capture the information regarding appearance and motion [14].
The size of the cuboids in the neighborhood of keypoints is determined by $\Delta_x, \Delta_y = 2q \sigma', \Delta_z = 2q \tau'$. Each cuboid is subdivided into a $n_x \times n_y \times n_t$ grid of cells. For each cell, we compute 4-bin HOG and 5-bin HOF. The computed histograms are normalized and concatenated to form the final descriptor which is similar to the famous SIFT descriptor. Parameter values of $q = 5, n_x, n_y = 3, n_t = 2$ are used in the experiments, which result in a descriptor with the dimensions of 162.

5.3.3 Codebook Size

After extracting features, we use them to obtain a codebook for each view. We apply k-means clustering on the training features for each view point individually using Euclidean distance. We use two codebooks, one of size $V$ and the other of size $2V$. According to Gehler and Nowozin [124], adding any feature (kernel), even uninformative and non-discriminative one, to the kernel weight optimization methods will not reduce the classification performance. In particular, when the added feature(kernel) is informative or discriminative, the classification performance will increase. Using two different sizes of vocabulary will enable us to model the actions with two different scales of detail. Here we empirically choose $V$ to be 75 for all the features. The features are subsequently quantized based on the vocabularies to form different histograms.

5.4 Efficient Combination of View-dependent Histograms

After computing histograms for each view and for each feature and codebook size, we obtain a number of different histograms representing an action video. Now in order to efficiently capture the similarities between different histograms we have to use the right measures of distance. Subsequently we use the distances in appropriate functions to form the similarities in a kernel matrix. In this chapter we use HIK distance within a linear function as well as $\chi^2$ distance within an exponential function in order to model actions in different spaces.
5.4.1 Computing Histogram Intersection Kernel

The Histogram Intersection Kernel (HIK) first introduced by Swain and Ballard [133] is a measure of similarity between histograms. HIK between histograms \( h \) and \( h' \) with \( d \) number of bins is defined as:

\[
k_{HIK}(h_a, h_b) = \sum_{i=1}^{d} \min(h_a(i), h_b(i)).
\]  

Barla et al. [134] showed that HIK is positive definite and hence, can be used as a kernel with SVM. Moreover they verified that image classification using SVM with HIK is effective. On the other hand, Maji et al. [135] proposed a method for accelerating the computation of the kernel which reduced its computational cost by a great extent. These facts motivate us to use HIK for our histograms.

5.4.2 Computing Radial Basis Function Kernel with \( \chi^2 \) Distance

Radial Basis Function (RBF) kernel nonlinearly maps the samples into a higher dimensional space. It is defined as [136]:

\[
k_{RBF}(h_a, h_b) = \exp(-\gamma D(h_a, h_b)), \gamma > 0,
\]  

where \( \gamma \) is the bandwidth parameter and \( D \) is an arbitrary distance metric. In this work we use \( \chi^2 \) distance as the distance metric, defined as:

\[
D_{\chi^2}(h_a, h_b) = \frac{1}{2} \sum_{i=1}^{d} \frac{(h_a(i) - h_b(i))^2}{h_a(i) + h_b(i)}.
\]  

\( \chi^2 \) distance is commonly used to compare histograms.

5.4.3 Learning an Efficient Combination of Kernels

The HIK and RBF kernels from different histograms need to be combined in an efficient way to acquire an optimized final kernel. The final kernel is used with SVM to classify the actions. Thus the binary SVM classifier will be in the form of [136]:

\[
F(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b\right),
\]  

84
Chapter 5. Efficient Combination of View-dependent Histograms for Multi-view Human Action Recognition

where \( \{x_i\}_{i=1}^N \) are the training sample videos and \( \{y_i\}_{i=1}^N \in \{-1, +1\} \) are the associated binary labels. \( \{\alpha_i\}_{i=1}^N \) and \( b \) are the parameters of SVM which are learned from the examples \(^1\). \( K \) is a linear combination of basis kernels \( K_m \) \cite{128}:

\[
K(x, x') = \sum_{m=1}^{M} w_m K_m(x, x'), 0 \leq w_m, \|w\|^2_2 \leq 1. \tag{5.7}
\]

\( \{K_m\}_{m=1}^M \) are computed from different histograms using either HIK or RBF kernel. \( w_m \) is the kernel weight which changes (scales) the influence of the kernel space associated with \( K_m \) and subsequently the corresponding histogram space. \( 0 \leq w_m \) is to ensure that the obtained matrices are positive-semidefinite. \( \|w\|^2_2 \leq 1 \) will ensure the convexity of the optimization function. We use 1-vs-1 classification scheme and choose different weights for each binary classifier. Choosing the optimum weights is important for the performance of the system, since some of the histograms (feature spaces) may be discriminative in differentiating between a pair of classes but non-informative for another pair. For instance, two actions may seem different from one viewpoint but similar from another view. Also a particular local feature may be more informative for distinguishing some pairs of actions but not others. Moreover, some actions are more efficiently represented using HIK kernels while other actions are more easily distinguished using RBF kernels. Due to these uncertainties, we use a learning method which estimates the weights from the data in an optimum way.

For every combination of local feature and codebook size, we incorporate only one instance of HIK kernel and four instances of RBF kernel. This is because HIK kernels are mixed linearly \( (w_1^{HIK} K_h + w_2^{HIK} K_h) \) but RBF kernels are combined with different bandwidths \( (w_1^{RBF} \exp(-\gamma_1 D_{x^2}) + w_2^{RBF} \exp(-\gamma_2 D_{x^2})) \) so they finally form different kernels. We use four different RBF kernels of bandwidths \( \gamma = 0.005, 0.05, 0.5, \) and 1.

We use a non-sparse multiple kernel learning algorithm \cite{128}. The common sparse kernel combination methods such as SILP \cite{137} or SimpleMKL \cite{138} are interpretable in the sense that only the relevant pieces of information or features (kernels) are extracted and used, the other features (kernels) are removed. Experimental results show that sparse methods do not perform much better than baseline methods using average weights \cite{139}.

\(^1\)note that \( \alpha_i \neq 0 \) only for support vectors
Thus we use a general $l_p$-norm multiple kernel learning where no feature is removed but all features participate and with different contributions. We empirically select the $l_2$-norm. Newton descent is used for optimization due to its faster performance compared to cutting planes [128].

We use the leave-one-person-out cross validation scheme. Thus each time we test on the sequences of one person and train on the sequences of the remaining persons. We iterate on all persons. The accuracy is computed as the average recognition rate over all repetitions. The multiple kernel learning algorithm gives both the optimum SVM parameters in equation (5.6) as well as the optimum kernel weights in equation (5.7).

## 5.5 Experimental Results

In this section we have carried out several experiments to show the efficiency and effectiveness of the proposed method for multi-view action recognition. We use the challenging IXMAS multi-view action dataset since it is the most used multi-view dataset publicly available. The dataset is detailed in the following subsection. Then we present and discuss about the results which is to the best of our knowledge the state-of-the-art for IXMAS dataset. Subsequently a variety of experiments are performed to study the effect of constitutive factors on the performance of the method. Finally we compare our method with other fusion techniques as well as recent results on IXMAS dataset. In all the experiments we fix the SVM cost parameter ($C$) to 1.

### 5.5.1 IXMAS Multi-view Action Dataset

The INRIA IXMAS dataset [12] is a challenging multi-view dataset for action recognition which is publicly available\(^2\). It comprises 14 daily-live actions, each performed three times by 12 actors. In order to test view-invariance, the actors freely change orientations in each performance without any information provided other than the labels. Example images of 11 actions are shown in figure 5.3. To be comparable, in our experiments, we have used the same 11 actions (figure 5.3) and 10 subjects which are used in [12] and [122]. The motions in IXMAS dataset are captured using five fire-wire cameras. Figure 5.4 shows example views from the five cameras for kick action. Similar to [12] and [122]

\(^2\)The dataset is accessible via http://4drepository.inrialpes.fr/public/viewgroup/6
Figure 5.3: Example images of the actions in IXMAS dataset.
Chapter 5. Efficient Combination of View-dependent Histograms for Multi-view Human Action Recognition

Figure 5.4: Example views from five cameras in IXMAS dataset for kick action.

we use leave-one-person-out cross validation. So in each cycle we train with the data of nine persons and test with the data of the remaining person. This procedure is repeated for all the 10 persons. The accuracy reported is the average of all the 10 runs.

5.5.2 Results and Analysis

In this part, we aim for the best performance using all the possible kernels. Thus we consider every combination of the constitutive factors, namely, 5 camera viewpoints \((C_1, C_2, C_3, C_4, C_5)\), 2 feature types (Separable Linear Filters and Space-time corner detectors), 2 codebook sizes \((V \text{ and } 2V)\) and 2 kernel types (HIK and RBF). Also we apply 4 RBF kernels instead of one. This gives a total number of 100 basic kernels to be linearly combined according to equation (5.7). The kernel weights are computed for each binary SVM separating a pair of classes. Figure 5.5 shows the confusion matrix for the IXMAS dataset. As seen most of the confusion occurs between check watch and cross arms or wave which have partially similar hand movements. Most other actions are perfectly discriminated. The overall recognition accuracy is 95.8% which is the state of the art in IXMAS dataset. The mean training time is 5340 seconds, while the mean testing time is only 5 seconds. So once the best combination of kernels are learned the evaluation of query action can be done pretty fast.

5.5.3 Viewpoint Analysis

In order to compare different camera viewpoints, we experiment with all the kernels extracted from each individual view of IXMAS. The result is shown in figure 5.6. As expected, the best accuracy is obtained with front (camera 2) and side (camera 4) views.
Figure 5.5: Confusion matrix for the best result achieved on IXMAS.

Figure 5.6: Accuracy for each view (camera) of IXMAS.
Figure 5.7: Best accuracy for combination of views in IXMAS.

The top view (camera 5) has the worst overall recognition rate since most of the movements are partially covered when observed from the top.

We also experiment combining different views together. Figure 5.7 shows the best results achieved using different number of cameras. Unsurprisingly, more views result in higher recognition accuracy.

5.5.4 Effect of Feature Type

In this part we study the performance of each local feature type individually. We combine all the basic kernels obtained with Separable Linear Filter (Dollar’s) and compare it to the combination of kernels using Space-time Corner Detector (Laptev’s). The result for using each feature-type separately as well as using both are illustrated in figure 5.8. As seen, Separable Linear Filter outperforms Space-time Corner Detector with a big margin. This is probably due to the much denser keypoints generated from the separable filter method as also mentioned by Dollar et. al [35]. Moreover the Gabor filter applied in time domain helps to capture the periodic motions more efficiently. The accuracy is more than 85% when only Dollar’s feature is used.

In order to compare our method with mixed-view fusion [105, 131] as explained in section 5.2, we also explored using histograms obtained from Dollar’s feature trained regardless of viewpoint using a single SVM classifier (with HIK kernel). We obtained 58.3% which is around 30% less than using our combination method using HIK kernels. The result reported in [105] is 82.8% which is obtained using all views without the top
view (four views) and based on a voting method. Also they have used 13 actions and 12 actors from IXMAS. The high misclassification rate is due to classifying different views with different appearances using the same model.

5.5.5 Effect of using more Codebooks

We have formed histograms based on two codebooks. These codebooks have been generated from the same set but have different sizes ($V$ and $2V$). Here we study how the performance is affected by adding one more codebook. Figure 5.9 shows the performance of 1 codebook versus 2 codebooks using all the basic kernels. As seen simply having two
sizes of vocabulary, will improve the performance. This is much easier and more computationally efficient than adding another feature. Having two different sizes of codebook will enable us to model the action in different scales of details. Adding more codebooks (more than two) may result in better accuracy but with higher computational cost.

5.5.6 Effect of Kernel Type

Here we explore the impact of choosing different kernel types (HIK kernel and RBF kernel with $\chi^2$ distance) as well as combining them on the performance. Therefore we have experimented with using only HIK kernels, only RBF kernels and combining both of them. Note that in this work we have used 4 RBF kernels with the same combinations. In section 5.4.3 we explained that we can use more than 1 RBF kernel with different bandwidths. The results are illustrated in figure 5.10. Using sufficient amount of training data, 1 RBF has almost the same performance as HIK, but as the training set gets smaller, HIK outperforms RBF. Using more number of RBF kernels will improve the results. But HIK still outperforms 4 RBF in a small training set. As anticipated combining everything will result in a higher accuracy.

Figure 5.10: Performance of each kernel type and combination of them.
Note that by using different types of kernels and combining them, we have increased the performance without having the trouble of extracting more information from the video. Actually we have represented the same data in different spaces with different metrics (histogram intersection and $\chi^2$). All of these models have its own pros and cons and none of them is absolutely better than the other one. Thus we combine them with adaptive weights which depend on the pairs of classes to optimize the performance.

### 5.5.7 Comparison of Different Fusion Methods

Our proposed method weights each basic kernel differently. In order to show that our weighting is efficient we compare our results with equal weighting of basic kernels. Thus to combine the basic kernels we simply average them to form the final kernel and then classify it with SVM. Since HIK and RBF kernels are from different spaces with different scales, we cannot simply add them. So we study them separately. Figure 5.11 shows the obtained results. According to the figure, our combination method outperforms simple averaging of kernels. In the case of RBF kernel, the difference is larger.
Table 5.1: Comparison of Recognition Accuracy on IXMAS dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ours</strong></td>
<td>95.8</td>
</tr>
<tr>
<td>Peng and Qian [122]</td>
<td>94.6</td>
</tr>
<tr>
<td>Weinland et al. [12]</td>
<td>93.3</td>
</tr>
<tr>
<td>Pehlivan and Duygulu [130]</td>
<td>90.9</td>
</tr>
<tr>
<td>Weinland et al. [140]</td>
<td>83.4</td>
</tr>
<tr>
<td>Liu et al. [141], Liu and Shah [105]</td>
<td>82.8</td>
</tr>
<tr>
<td>Weinland et al. [142]</td>
<td>81.3</td>
</tr>
<tr>
<td>Lv and Nevatia [143]</td>
<td>80.6</td>
</tr>
<tr>
<td>Yan et al. [144]</td>
<td>78.0</td>
</tr>
<tr>
<td>Liu et al. [145]</td>
<td>73.8</td>
</tr>
<tr>
<td>Junejo et al. [146]</td>
<td>71.2</td>
</tr>
</tbody>
</table>

5.5.8 Comparison with Other Methods on IXMAS Dataset

To better study the efficiency of the proposed method, we have compared our result with the recent results of action recognition on IXMAS dataset in Table 5.1. As seen our proposed method outperforms the other approaches. Similar to the methods of Peng and Qian [122], Weinland et al. [12] and Pehlivan and Duygulu [130], our method also uses multiple views for recognition. However the other approaches that are based on single view recognition have lower results by a margin of 7.5%. Among the multiview approaches, our method performs better than the methods using 3D visual hull [12, 122, 130].

5.6 Conclusion

In this chapter we have proposed a novel method for combining the information from multiple viewpoints. Based on the bag-of-words framework, different histograms are constructed from each viewpoint by changing the codebook size. The similarity between histograms are represented by Histogram Intersection as well as \( \chi^2 \) distance in an exponential function. All the basic kernels generated are combined linearly with proper weights obtained by optimization. The final kernel is used with SVM. We have performed plenty of experiments to study the influence of each constitutive factor and verify the
efficiency of the method. State of the art result is achieved on the challenging multi-view IXMAS dataset.
Chapter 6

Conclusions

In this thesis we have proposed some novel methods for automatic recognition of human actions in videos. We introduced an efficient embedding method for learning the manifold of human actions. Next, we proposed a novel approach for introducing semantic relations into the bag of words framework by embedding the visual words into a concept space. Finally we present a new method for efficient combination of the information from multiple viewpoints. In this chapter we summarize the contributions presented in this dissertation and also look forward to the directions for future research in this area.

6.1 Summary of Contributions

(i) Chapter 3 introduces a novel embedding method for action recognition in the sequence recognition framework. This embedding which we call pose-based discriminant embedding (PDE) gives the optimum mapping based on the spatio-temporal correlation distance (SCD). Specifically, the proposed embedding maximizes sum of distances between inter-class points while trying to minimize sum of distances between intra-class sequences. Action sequences are modeled by special poses chosen equidistantly within one semantic period of action. We project the sequences of actions using the computed embedding. Action recognition is done by comparing the projected sequences in the low-dimensional subspace using SCD or Hausdorff Distance in the nearest neighbor framework. We have verified that the proposed embedding outperforms other dimension reduction techniques and performs faster by choosing less number of poses. Moreover it is robust to additive noise and tolerant to occlusion, deformation and change in viewpoint to a great extent.
(ii) Chapter 4 proposes a novel approach for introducing semantic relations into bag of words approach. Latent Semantic models such as LSA or pLSA as well as canonical correlation analysis (CCA) has been used to obtain a subspace in which synonym words are distributed close to each other. We refer to this space as the concept space. The extracted features are clustered using k-means to form a codebook. Then these visual words are embedded into the concept space and further clustered using k-means to construct semantically meaningful clusters which are used as the semantic visual vocabulary. Exploiting the obtained semantic visual codebook the features will form more discriminative histograms. Experiments have verified the efficiency of the proposed approach.

(iii) In Chapter 5, we present a new method for combining data from multiple viewpoints. Spatio-temporal features are extracted from each viewpoint and exploited in bag-of-words framework to form histograms. Two different sizes of codebook is being used. The similarity between the histograms are captured by Histogram Intersection Kernel (HIK) as well as RBF kernel with $\chi^2$ distance. The final kernel used is a linear combination of all the basic kernels generated by varying the viewpoints as well as different feature types, codebook sizes and kernel types. We use a one versus one binary classification scheme for our multi-class problem. The optimum kernel weights for each binary SVM classifier is found using an optimization process. We have obtained the state of the art accuracy on the challenging IXMAS multi-view action dataset.

6.2 Future Research

Here we discuss some of the potential future destinations relevant to human action recognition.

(i) As we have seen in chapter 3, representing actions in the manifold learning framework have proved to be effective in recognizing actions. A drawback of most of the existing well-known subspace learning methods including PCA, LDA, LPP as well as our proposed embedding, PDE, is that they are computationally expensive
in the sense that they need to compute eigen decompositions of dense matrices. Cai et al. [147] have proposed an embedding called Spectral Regression (SR) for efficient regularized subspace learning. It combines spectral graph theory and regression to perform efficient regularized embedding. SR only needs to solve some regularized least square problems instead of computationally expensive computations of eigen-decomposition which is an enormous save in time and memory (from cubic complexity to linear complexity). Furthermore, most of the previous embedding techniques can be restated in SR framework cutting down the computations. Experimental results on face recognition application have proved the efficiency of SR compared to previous embeddings. This motivates us to apply SR in action recognition framework.

Another direction which attracts researchers is using the distance between subspaces in classification. One of the most useful measures of distances known so far is principal angles, which have been used for face and action recognition [30, 83]. But this distance is defined between linear subspaces. However the wide variations in intra-class examples in the applications make the linearity a limiting condition. So we intend to look at more efficient distances which are defined between nonlinear subspaces as well. For example, Boosted manifold principal angles [148] is an important work which extends the concept of principal angles between linear subspaces to manifolds with arbitrary nonlinearities. It is showed in their work how boosting can be used for application-optimal principal angle fusion, which motivates the use of the method in action recognition task.

(ii) In Chapter 4, we presented a novel approach for exploiting semantic relations in action recognition by embedding visual words into concept space. One of the important parameters of this method is the dimension of the concept space or the number of latent topics. As mentioned, one of the significant advantages of this method is that the number of latent topics does not require to be equal to the number of categories. Particularly increasing the number of topics from the number of categories will enable us to model the concepts in more details and consequently increase the performance. However too many latent topics will decrease the performance since
the semantic units will no longer be meaningful. Thus there is an optimum number
of topics which lead to the best performance. In chapter 4, we found this optimum
number by experimenting different values. However any method to estimate this
parameter automatically from the data will be pleasing and will lead to a more
practical semantic approach.

Furthermore in chapter 4, by using the BOW framework we ignore the location
of interest points within the video which may be useful for recognition. Finding a
method to include spatial information efficiently might increase the overall perfor-
mance.

(iii) All the methods proposed in this thesis are based on offline learning algorithms for
classification. Thus there should be a sufficiently large dataset of labeled samples
to train the system offline. Moreover in order for the system to perform well
these training samples should include sufficient samples from all types of intra-class
variances. However, collecting large number of labeled samples require much time
and effort and is not always practical. While unlabeled data are easily available from
public surveillance camera streams and does not require human labor. Therefore
using semi-supervised learning algorithms which use both labeled and non-labeled
data simultaneously seems so beneficial. For instance, co-training [149, 150] as
a semi-supervised algorithm can be used to learn a weak classifier from an initial
small labeled training set. Then the learned classifier is used to predict labels for the
unlabeled instances. By finding the instances whose labels are the most confident
and adding them to the training set and re-learning the classifier, the performance
will increase. This process can be repeated for several iterations within an online
learning framework.

(iv) In this dissertation we focused on the so called action recognition task. But in real-
life we are also facing complex activities including any kind of interactions including
human-object interactions as well as human-human interactions. Actions can be
considered as building blocks for these higher level activities. So human action
recognition is the starting point for studying real-life activities. Complex activity
recognition systems are usually hierarchical composed of object recognition engines
as well as action recognition in the lower level and reasoning module in the higher level [8]. Techniques in language modeling can be adopted for this application [151].
Publications

(i) B. Saghafi and D. Rajan, “Efficient Combination of View-dependent Histograms for Multi-view Human Action Recognition,” submitted to *Image and Vision Computing*.


References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


110
REFERENCES


REFERENCES


REFERENCES


REFERENCES


